(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第5765416号
(P5765416)

(45)発行日　平成27年8月19日(2015.8.19)　(24)登録日　平成27年6月26日(2015.6.26)

(51)Int.Cl.　F I

G06F 12/00 (2006.01)	G06F	12/00	533J
G06F 3/06 (2006.01)	G06F	12/00	514E
	G06F	12/00	545A
	G06F	3/06	304F

請求項の数 28　(全 46 頁)

(21)出願番号　特願2013-503590 (P2013-503590)
(86)(22)出願日　平成24年3月8日(2012.3.8)
(86)国際出願番号　PCT/JP2012/055917
(87)国際公開番号　WO2012/121316
(87)国際公開日　平成24年9月13日(2012.9.13)
審査請求日　平成25年9月6日(2013.9.6)
(31)優先権主張番号　特願2011-50151 (P2011-50151)
(32)優先日　平成23年3月8日(2011.3.8)
(33)優先権主張国　日本国(JP)

(73)特許権者　000004237
日本電気株式会社
東京都港区芝五丁目７番１号
(74)代理人　100080816
弁理士　加藤　朝道
(72)発明者　菅　真樹
東京都港区芝五丁目７番１号　日本電気株式会社内
(72)発明者　鳥居　隆史
東京都港区芝五丁目７番１号　日本電気株式会社内

審査官　池田　聡史

最終頁に続く

(54)【発明の名称】分散ストレージシステムおよび方法

(57)【特許請求の範囲】
【請求項１】
それぞれがデータ格納部を備え、ネットワーク結合される複数のデータノードを備え、データ複製先のデータノードが、前記データノード間で、論理的には同一であるが、物理的には異なるデータ構造をそれぞれの前記データ格納部に保持する、少なくとも二つのデータノードを含み、前記複製先のデータノードは、目的のデータ構造への変換を複製データの受付とは非同期で行い、
データの識別子に対応して、前記データの格納先のデータノードと、データ構造の種類を特定するデータ構造管理情報を記憶管理する構造情報管理手段を備え、
前記データノードは、
アクセス要求に基づき、データの更新処理を行う場合に、受け付けたデータを、一旦、中間データ保持構造に保持して更新に対する応答を返すアクセス手段と、
設定された更新契機に応答して、前記中間データ保持構造に保持されるデータを、前記データ構造管理情報で指定されたデータ構造に非同期で変換して前記データ格納部に格納するデータ構造変換手段と、
を備えた、分散ストレージシステム。
【請求項２】
前記構造情報管理手段は、前記中間データ保持構造に保持されるデータが目的のデータ構造に変換して格納されるまでの時間情報である更新契機情報を、前記データ構造の種類に対応させて保持する、請求項１記載の分散ストレージシステム。

【請求項３】
前記構造情報管理手段は、前記データの識別子に対応して、１つ又は複数のデータ配置先のデータノード情報を特定するデータ配置特定情報をさらに記憶管理し、
前記更新処理に対応してアクセス先のデータノードを、前記データ構造管理情報と前記データ配置特定情報より選択するクライアント機能実現手段を備えた、請求項２記載の分散ストレージシステム。
【請求項４】
予め定められたテーブル単位でデータの配置先のデータノード、配置先でのデータ構造、データ分割を可変に制御する手段を備えた請求項１記載の分散ストレージシステム。
【請求項５】
データが配置されるデータノードを、コンシステント・ハッシングで求める手段を備えた、請求項１乃至４のいずれか１項に記載の分散ストレージシステム。
【請求項６】
それぞれがデータ格納部を備え、ネットワーク結合される複数のデータノードを備え、データ複製先のデータノードが、前記データノード間で、論理的には同一であるが、物理的には異なるデータ構造をそれぞれの前記データ格納部に保持する、少なくとも二つのデータノードを含み、
格納対象のデータを識別する識別子であるテーブル識別子に対応させて、複製を特定するレプリカ識別子と、前記レプリカ識別子に対応したデータ構造の種類を特定するデータ構造情報と、指定されたデータ構造に変換して格納されるまでの時間情報である更新契機情報と、を、前記データ構造の種類の数に対応させて備えたデータ構造管理情報と、
前記テーブル識別子に対応して、前記レプリカ識別子と、前記レプリカ識別子に対応した１つ又は複数のデータ配置先のデータノード情報と、を備えたデータ配置特定情報と、
を記憶管理する構造情報保持部を有する構造情報管理装置と、
前記データ構造管理情報と前記データ配置特定情報とを参照して、更新処理及び参照処理のアクセス先を特定するデータアクセス部を備えたクライアント機能実現部と、
それぞれが前記データ格納部を備え、前記構造情報管理装置と前記クライアント機能実現部とに接続される複数の前記データノードと、
を備え、
前記データノードは、
前記クライアント機能実現部からのアクセス要求に基づき、更新処理を行う場合に、中間データ保持構造にデータを保持して前記クライアント機能実現部に応答を返すデータ管理・処理部と、
前記データ構造管理情報を参照し、指定された更新契機に応答して、前記中間データ保持構造に保持されるデータを、前記データ構造管理情報で指定されたデータ構造に変換する処理を行うデータ構造変換部と、
を備えることを特徴とする、分散ストレージシステム。
【請求項７】
予め定められたテーブル単位でデータの配置先のデータノード、配置先でのデータ構造、データ分割を可変に制御する手段を備えた請求項６記載の分散ストレージシステム。
【請求項８】
前記中間データ保持構造は、指定された目的のデータ構造としてデータが前記データ格納部に格納されるまでの間、前記データを保持する、請求項６記載の分散ストレージシステム。
【請求項９】
前記クライアント機能実現部が、前記更新処理又は前記参照処理の内容に応じてアクセス先のデータノードを、前記データ構造管理情報と前記データ配置特定情報より選択する、請求項６記載の分散ストレージシステム。
【請求項１０】
前記クライアント機能実現部は、前記構造情報管理装置の前記構造情報保持部に保持さ

れている前記データ配置特定情報、又は、前記構造情報保持部に保持される情報をキャッシュする構造情報キャッシュ保持部に保持されているデータ配置特定情報を取得し、データ配置先のデータノードに対して、アクセス命令を発行する、請求項６記載の分散ストレージシステム。

【請求項１１】

前記データノードは、アクセス受付部、アクセス処理部、データ構造変換部を備え、

前記データノードの前記データ格納部は、構造別データ格納部を備え、

前記アクセス受付部は、前記クライアント機能実現部からの更新要求を受け付け、前記データ配置特定情報においてレプリカ識別子に対応して指定されているデータノードに対して更新要求を転送し、

前記データノードの前記アクセス処理部は、受け取った更新要求の処理を行い、前記データ構造管理情報の情報を参照して更新処理を実行し、その際、前記データ構造管理情報の情報から、前記データノードに対する前記更新契機が零の場合、更新データを、前記データ構造管理情報に指定されるデータ構造に変換して前記構造別データ格納部を更新し、

前記更新契機が零でない場合、前記中間データ保持構造に、一旦、更新データを書き込み、処理完了を応答し、

前記アクセス受付部は、前記アクセス処理部からの完了通知と、レプリカ先のデータノードの完了通知を受けると、前記クライアント機能実現部に対して応答し、

前記データ構造変換部は、前記中間データ保持構造のデータを、前記データ構造管理情報に指定されているデータ構造に変換し変換先の前記構造別データ格納部に格納する、請求項６又は１０記載の分散ストレージシステム。

【請求項１２】

前記クライアント機能実現部は、参照系アクセスの場合、データノードに対して行われるデータアクセスに適しているデータ構造を選択し、レプリカ識別子を特定した後、アクセスすべきデータノードを算出し、選択されたデータノードに対してアクセス要求を発行し前記データノードからアクセス処理結果を受け取る、請求項６記載の分散ストレージシステム。

【請求項１３】

前記クライアント機能実現部が、前記データノード内に配設されている、請求項６記載の分散ストレージシステム。

【請求項１４】

前記クライアント機能実現部が、前記構造情報保持部に保持される情報をキャッシュする構造情報キャッシュ保持部を備えた請求項１３記載の分散ストレージシステム。

【請求項１５】

前記クライアント機能実現部の前記構造情報キャッシュ保持部の構造情報と、前記構造情報管理装置の前記構造情報保持部に保持される構造情報を同期させる構造情報同期部を備えた請求項１４記載の分散ストレージシステム。

【請求項１６】

前記データ構造管理情報が、データを複数のデータノードに分割して格納する分割数であるパーティション数をレプリカ識別子に対応して備え、

前記データ配置特定情報は、前記データ構造管理情報においてパーティション数が２以上に対応するレプリカ識別子に対応した配置ノードとして、複数のデータノードを含み、

アクセス要求を受けた前記データノードのアクセス受付部は、パーティショニングされたデータの配置先が複数のデータノードにまたがる場合に、前記複数のデータノードを構成する他のデータノードのアクセス処理部にアクセス要求を発行する、請求項６記載の分散ストレージシステム。

【請求項１７】

アクセス要求を受けた前記データノードの前記データ構造変換部は、前記更新契機が零のとき、他のデータノードの前記データ構造変換部に対してアクセス要求を発行する、請求項６又は１１記載の分散ストレージシステム。

【請求項１８】
アクセス要求の履歴を記録する履歴記録部と、
前記履歴記録部の履歴情報を用いてデータ構造の変換を行うか否かを判定する変更判定部と、
を備えた請求項６記載の分散ストレージシステム。
【請求項１９】
前記変更判定部は、データ構造の変換が必要と判定した場合、前記構造情報管理装置の構造情報変更部に変換要求を出力し、
前記構造情報管理装置の前記構造情報変更部は、前記構造情報保持部の情報を変更し、前記データノードの前記データ構造変換部に変換要求を出力し、
前記データノードの前記データ構造変換部は前記データノードの前記データ格納部に保持されるデータ構造の変換を行う、請求項１８記載の分散ストレージシステム。
【請求項２０】
それぞれがデータ格納部を備え、ネットワーク結合される複数のデータノードを備えたシステムでの分散ストレージ方法であって、
データ複製先のデータノードの少なくとも二つのデータノードが、前記データノード間で、論理的には同一であるが、物理的には異なるデータ構造をそれぞれの前記データ格納部に保持し、前記複製先のデータノードは、目的のデータ構造への変換を複製データの受付とは非同期で行い、
データの識別子に対応して、前記データの格納先のデータノードと、データ構造の種類を特定するデータ構造管理情報を構造情報管理手段で記憶管理し、
前記データノードは、
アクセス要求に基づき、データの更新処理を行う場合に、受け付けたデータを、一旦、中間データ保持構造に保持して、更新に対する応答を返し、
データの更新契機に応答して、前記中間データ保持構造に保持されるデータを、前記データ構造管理情報で指定されたデータ構造に非同期で変換し前記データ格納部に格納する、分散ストレージ方法。
【請求項２１】
前記構造情報管理手段は、前記中間データ保持構造に保持されるデータが目的のデータ構造に変換して格納されるまでの時間情報である更新契機情報を、前記データ構造の種類に対応させて保持する、請求項２０記載の分散ストレージ方法。
【請求項２２】
前記構造情報管理手段では、前記データの識別子に対応して、１つ又は複数のデータ配置先のデータノード情報を特定するデータ配置特定情報をさらに記憶管理し、
前記更新処理に対応してアクセス先のデータノードを、前記データ構造管理情報と前記データ配置特定情報より選択する、請求項２１記載の分散ストレージ方法。
【請求項２３】
予め定められたテーブル単位でデータの配置先のデータノード、配置先でのデータ構造、データ分割を可変に制御する、請求項２０記載の分散ストレージ方法。
【請求項２４】
データが配置されるデータノードをコンシステント・ハッシングで求める、請求項２０乃至２３のいずれか１項に記載の分散ストレージ方法。
【請求項２５】
それぞれがデータ格納部を備え、ネットワーク結合される複数のデータノードを備えたシステムでの分散ストレージ方法であって、
データ複製先のデータノードの少なくとも二つのデータノードが、前記データノード間で、論理的には同一であるが、物理的には異なるデータ構造をそれぞれの前記データ格納部に保持し、
データ更新時に行われるデータの複製において、前記複製先のデータノードでは、更新対象のデータを、それぞれ、指定された目的のデータ構造に変換して前記データ格納部に

格納し、その際、前記データノードは、更新対象のデータを一旦、中間構造を保持して前記更新に対する応答を返し、更新要求とは非同期で、目的のデータ構造に変換して格納し、

格納対象のデータを識別する識別子であるテーブル識別子に対応させて、複製を特定するレプリカ識別子と、前記レプリカ識別子に対応したデータ構造の種類を特定するデータ構造情報と、指定されたデータ構造に変換して格納されるまでの時間情報である更新契機情報と、を前記データ構造の種類の数に対応させて備えたデータ構造管理情報と、

前記テーブル識別子に対応して、前記レプリカ識別子と、前記レプリカ識別子に対応した１つ又は複数のデータ配置先のデータノード情報と、を備えたデータ配置特定情報と、

を含む構造情報を構造情報管理部で記憶管理し、

クライアント側では、前記データ構造管理情報と前記データ配置特定情報を参照して、更新処理及び参照処理のアクセス先を特定し、

前記データノードは、

前記クライアント側からのアクセス要求に基き、更新処理を行う場合に、中間データ保持構造にデータを保持して前記クライアントに応答を返し、

前記データ構造管理情報を参照し、指定された更新契機に応じて、前記中間データ保持構造から指定されたデータ構造に変換する、

ことを特徴とする分散ストレージ方法。

【請求項２６】

前記データ構造管理情報が、データを複数のデータノードに分割して格納する分割数であるパーティション数を、レプリカ識別子に対応して備え、

前記データ配置特定情報は、前記データ構造管理情報においてパーティション数が２以上に対応するレプリカ識別子に対応した配置ノードとして、複数のデータノードを含み、

アクセス要求を受けた前記データノードでは、パーティショニングされたデータの配置先が複数のデータノードにまたがる場合に、前記複数のデータノードを構成する他のデータノードに対してアクセス要求を発行する、請求項２５記載の分散ストレージ方法。

【請求項２７】

アクセス要求の履歴を記録する履歴記録部での履歴情報を用いて、データ構造の変換を行うか否かを判定し、変換が必要な場合、前記構造情報を変換し、さらに前記データノードのデータ構造を変換する、請求項２５記載の分散ストレージ方法。

【請求項２８】

データ格納部を備え、他のデータノードとネットワーク結合され、複数のデータノードが分散ストレージシステムを構成し、

更新対象のデータを複数のデータノードに複製する場合、前記データに関して、少なくとも一つの他のデータノードとの間で、論理的には同一であるが、物理的には異なるデータ構造を前記データ格納部に保持し、目的のデータ構造への変換を複製データの受付とは非同期で行うデータノード装置であって、

アクセス要求に基づき、データの更新処理を行う場合に、受け付けたデータを、一旦、中間データ保持構造に保持して更新に対する応答を返すアクセス手段と、

データの識別子に対応して、前記データの格納先のデータノードと、データ構造の種類を特定し、前記データの更新契機を設定するデータ構造管理情報を記憶管理する構造情報管理装置からの前記データ構造管理情報に基づき、設定された更新契機に応答して、前記中間データ保持構造に保持されるデータを、前記データ構造管理情報で指定されたデータ構造に非同期で変換して前記データ格納部に格納するデータ構造変換手段と、

を備えたデータノード装置。

【発明の詳細な説明】

【技術分野】

【０００１】

（関連出願についての記載）

本発明は、日本国特許出願：特願２０１１－０５０１５１号（２０１１年３月８日出願

）の優先権主張に基づくものであり、同出願の全記載内容は引用をもって本書に組み込み記載されているものとする。

【０００２】

本発明は、分散ストレージに関し、特に、データ構造の制御が可能な分散ストレージシステム、および方法と装置に関する。

【背景技術】

【０００３】

＜分散ストレージシステム＞

複数の計算機（データノード、あるいは単に「ノード」ともいう）をネットワーク結合し、各計算機のデータ格納部（ＨＤＤ（Ｈａｒｄ　Ｄｉｓｋ　Ｄｒｉｖｅ）やメモリ等）にデータを格納して利用するシステムを実現する分散ストレージシステム（Ｄｉｓｔｒｉｂｕｔｅｄ　Ｓｔｏｒａｇｅ　Ｓｙｓｔｅｍ）が利用されている。

【０００４】

一般的な分散ストレージ技術では、

・データをどの計算機（ノード）に配置するか、

・処理をどの計算機（ノード）で行うか、

といった判断をソフトウェアや特別な専用ハードウェアにより実現し、システムの状態に対してその動作を動的に変更することでシステム内のリソース使用量を調整し、システム利用者（クライアント計算機）に対する性能を向上している。

【０００５】

分散ストレージシステムにおいては、データが複数のノードに分散しているため、データにアクセスしようとするクライアントは、まず、データを持っているノードがどれであるかを知る必要がある。また当該データを持つノードが複数ある場合、どのノード（一つ以上）にアクセスするかを知る必要がある。

【０００６】

分散ストレージシステムでは、一般に、ファイル管理として、ファイル本体と、当該ファイルのメタデータ（ファイルの格納場所、ファイルサイズ、オウナー等）を別々に保存する方式が用いられている。

【０００７】

＜メタサーバ方式＞

分散ストレージシステムにおいて、クライアントがデータを保持しているノードを知るための技術の一つとしてメタサーバ方式が知られている。メタサーバ方式では、データの位置情報を管理する、一つ又は複数（ただし、少ない数）の計算機により構成されたメタサーバを設ける。

【０００８】

しかしながら、メタサーバ方式では、分散ストレージシステムの構成の大規模化に伴って、データを格納しているノードの位置を検出する処理を行うメタサーバの処理性能が足りず（メタサーバ１台当りで管理するノード数が膨大となり、該メタサーバの処理性能が追いつかない）、導入したメタサーバが、かえってアクセス性能上のボトルネックになる、という問題がある。

【０００９】

＜分散ＫＶＳ＞

データを保持しているノードの位置を知るための別の手法（技術）として、分散関数（例えばハッシュ関数）を用いてデータの位置を求めるものがある。この種の手法は、例えば分散ＫＶＳ（Ｋｅｙ　Ｖａｌｕｅ　Ｓｔｏｒｅ：キー・バリュー・ストア）と呼ばれている。

【００１０】

分散ＫＶＳでは、全てのクライアントで、分散関数と、システムに参加しているノードのリスト（ノードリスト）とを共有する。

【００１１】

また、格納データは、固定長あるいは任意長のデータ断片（Ｖａｌｕｅ）に分かれている。各データ断片（Ｖａｌｕｅ）毎に一意に特定可能な識別子（Ｋｅｙ）が付与され、（Ｋｅｙ、Ｖａｌｕｅ）のペアで保存される。例えばキーの値に応じて保存先のノード（サーバ）を変えることで、複数のノードにデータを分散保存することが可能となる。

【００１２】

各クライアントは、データにアクセスする際、キーを分散関数の入力値とし、分散関数の出力値とノードリストとを基に、データを格納しているノードの位置を算術的に求める。

【００１３】

クライアント間で共有する情報のうち、分散関数は、基本的に、時間が経過しても変化しない（時不変）。一方、ノードリストの内容は、ノードの故障や追加に伴い、随時、変更される。このため、クライアントは、それらの情報に対して任意の方法でアクセス出来る必要がある。

【００１４】

＜レプリケーション＞

分散ストレージシステムにおいては、可用性（Ａｖａｉｌａｂｉｌｉｔｙ：システムが連続して動作できる能力）確保のために、データの複製を複数ノードで保持し、データの複製を、負荷分散に活用することが一般的に行われている。

【００１５】

なお、作成するデータの複製を用いて負荷分散を実現する技術が特許文献１に記載されている。

【００１６】

本件に関して行われた先行文献サーチの結果サーチされた特許文献２には、サーバが情報構造定義部で情報構造定義体を定義し、登録用クライアントは情報構造定義体によりデータベースを構築し、データベースアクセスツールを生成し、このツールを用いてデータベースに情報を登録する構成が開示されている。また特許文献３には、分散型ストレージシステムにおいて、各複製がそれぞれ固有のロケータ値を介してアクセス可能なオブジェクトの複製を保存するストレージノードと、各オブジェクトに対するそれぞれのキーマップエントリを保存するキーマップインスタンスを含み、所定のオブジェクトについてはそれぞれのキーマップエントリは、オブジェクトの複製と、対応するキー値、各ロケータを含む構成が開示されている。

【先行技術文献】

【特許文献】

【００１７】

【特許文献１】特開２００６－１２００５号公報（特許第４５２８０３９号）
【特許文献２】特開平１１－１９５０４４号公報（特許第３９１１８１０号）
【特許文献３】特表２００９－５２２６５９号公報

【発明の概要】

【発明が解決しようとする課題】

【００１８】

以下に関連技術の分析を与える。

【００１９】

関連技術の分散ストレージシステムでは、可用性保持のため、複製データを複数のノードで同一の物理構造で保持している。これにより、アクセス応答性能と可用性保証を実現している。しかしながら、複製データを同一の物理構造で保持しているため、データの利用形態の特性が異なるアプリケーション等に対しては、別のデータ構造への変換、及び別のデータ構造を保持するためのストレージを用意しなければならない。

【００２０】

したがって、本発明の目的は、分散ストレージにおけるデータ複製において、可用性を確保するとともに、ストレージの利用効率の低下の回避、応答性能の低下の回避の少なく

とも１つを可能とする、分散ストレージシステムと方法を提供することにある。
【課題を解決するための手段】
【００２１】
本発明によれば、上記課題の少なくとも１つの解決を図るため、特に制限されるものでないが、概略以下の構成とされる。
【００２２】
本発明によれば、それぞれがデータ格納部を備え、ネットワーク結合される複数のデータノードを備え、データ複製先のデータノードが、前記データノード間で、論理的には同一であるが、物理的には異なるデータ構造をそれぞれの前記データ格納部に保持する、少なくとも二つのデータノードを含む分散ストレージシステムが提供される。本発明によれば、分散ストレージシステムを構成するデータノード装置として、他のデータノードとネットワーク結合され、更新対象のデータを複数のデータノードに複製する場合、前記データに関して、少なくとも一つの他のデータノードとの間で、論理的には同一であるが、物理的には異なるデータ構造を前記データ格納部に保持するデータノード装置が提供される。
【００２３】
本発明によれば、それぞれがデータ格納部を備え、ネットワーク結合される複数のデータノードを備えたシステムでの分散ストレージ方法であって、前記複数のデータノードの少なくとも二つのデータノードは、前記データノード間で論理的には同一であるが、物理的には異なる複数種のデータ構造の複製をそれぞれの前記データ格納部に保持する、分散ストレージ方法が提供される。
【００２４】
いくつかの実施形態によれば、前記複数のデータノードにおいて、ダーゲットのデータ構造への変換をデータ更新要求とは非同期で行うようにしてもよい。あるいは、いくつかの実施形態によれば、前記データノードにおいて中間データ保持構造に受信データを保持し、前記更新要求の応答を返し、前記中間データ保持構造に保持されるデータ構造をターゲットのデータ構造に非同期で変換する。あるいは、いくつかの実施形態によれば、予め定められたテーブル単位でデータ配置先、配置先のデータ構造、データ分割を可変に制御する。
【発明の効果】
【００２５】
本発明によれば、分散ストレージにおけるデータ複製において、可用性を確保するとともに、ストレージの利用効率の低下の回避、応答性能の低下を回避の少なくとも１つ可能としている。
【図面の簡単な説明】
【００２６】
【図１】本発明の第１の例示的な実施の形態のシステム構成を示す図である。
【図２】本発明の第１の例示的な実施の形態を説明する図である。
【図３】本発明の第１の例示的な実施形態を説明する図である。
【図４】本発明の第１の例示的な実施形態のデータノードの構成例を示す図である。
【図５】本発明の第１の例示的な実施形態におけるデータ構造管理情報９２１を模式的に示す図である。
【図６】本発明の第１の例示的な実施形態におけるテーブルのデータ保持構造を説明する図である。
【図７】本発明の第１の例示的な実施形態におけるテーブルのデータ保持、非同期更新を模式的に説明する図である。
【図８】本発明の第１の例示的な実施形態におけるデータ配置特定情報９２２の例を示す図である。
【図９】本発明の第１の例示的な実施形態におけるＷｒｉｔｅ処理の動作シーケンスを説明する図（１）である。

【図１０】本発明の第１の例示的な実施形態におけるＷｒｉｔｅ処理の動作シーケンスを説明する図（２）である。
【図１１】本発明の第１の例示的な実施形態におけるＲＥＡＤ系処理の動作シーケンスを説明する図である。
【図１２】本発明の第１の例示的な実施形態におけるクライアント実現手段６１におけるアクセス処理の動作を説明するフローチャートである。
【図１３】本発明の第１の例示的な実施形態におけるデータノードにおけるアクセス処理の動作を説明するフローチャートである。
【図１４】本発明の第１の例示的な実施形態におけるデータ変換処理を説明するフローチャートである。
【図１５】本発明の第２の例示的な実施形態のデータ構造管理情報９２１を模式的に示す図である。
【図１６】本発明の第２の例示的な実施形態におけるデータ配置特定情報９２２の例を示す図である。
【図１７】本発明の第２の例示的な実施形態のデータノードの構成例を示す図である。
【図１８】本発明の第３の例示的な実施形態のデータノードの構成例を示す図である。
【図１９】本発明の第３の例示的な実施形態の全体の制御フローを説明するフローチャートである。
【図２０】本発明の第３の例示的な実施形態のデータ構造変換処理を説明するフローチャートである。
【図２１】本発明の第３の例示的な実施形態の変換処理を説明する図である。
【図２２】本発明の第３の例示的な実施形態のパーティショニング数の変更処理を説明するフローチャートである。
【図２３】本発明の第３の例示的な実施形態のパーティショニング数変更時の動作を説明するフローチャートである。
【図２４】本発明の第３の例示的な実施形態における分散テーブルのデータ配置を説明する図である。
【図２５】本発明の第３の例示的な実施形態における構造情報保持部９２を説明する図である。
【図２６】本発明の第４の例示的な実施形態のコンシステント・ハッシング分割配置を説明する図である。
【図２７】本発明の第４の例示的な実施形態の情報記録形態を説明する図である。
【図２８】本発明の第４の例示的な実施形態におけるカラムベースのコンシステント・ハッシング分割配置を説明する図である。
【図２９】本発明の第４の例示的な実施形態において１カラムをパーティショニングした場合のコンシステント・ハッシング分割配置を説明する図である。
【発明を実施するための形態】
【００２７】
本発明の好ましい態様（Ｐｒｅｆｅｒｒｅｄ　Ｍｏｄｅｓ）の一つによれば、複数種類のデータ構造を持ち、データ配置ノード（「データノード」という）間で論理的には同一であるが、物理的には異なる構造の複製（レプリカ）を保持する。本発明においては、書き込み（更新）要求とは非同期に行われるデータ構造変換の適用契機を制御可能としている。本発明においては、Ｗｒｉｔｅの応答特性を優先した構造を中間構造（中間データ保持構造）を備え、該中間構造に保持されるデータ構造をターゲットとなるデータ構造に非同期に変換する。
【００２８】
本発明の好ましい態様においては、制御パラメータを変更するインタフェースを持つ。アクセス負荷に応じて制御パラメータを変更する。あるいは、処理負荷が増えたら、パーティショニング粒度を小さくする等の制御が行われる。
【００２９】

本発明の好ましい態様によれば、複数種類のデータ構造を持つことが可能となるキー・バリュー・ストア（Ｋｅｙ　Ｖａｌｕｅ　Ｓｔｏｒｅ）を実現可能としている。本発明の好ましい態様によれば、論理的には同一内容であるが、物理的には異なるデータ構造の複製（レプリカ）を持つ。この結果、

・異なる種類のアクセス負荷に対して高速に対応可能とし、

・可用性保持のための複製（レプリカ）を他の用途に利用可能とし、データ容量の効率利用を可能としている。

【００３０】

本発明の好ましい態様において、データ送信元から該データを受け取る側のデータノードでは、受信データを複製に同期して直ちにターゲット構造に変換する代りに、中間構造形式で保持し、ターゲット構造への変換を非同期で行うようにしてもよい。例えばＷｒｉｔｅ要求に対してデータをバッファに保持して直ちに応答を返す等、アクセス要求に対する応答特性を優先した中間構造を備え、中間構造に保持されたデータ構造を、非同期でターゲット構造を変換することにより、データ構造の変換処理によって生じる、アクセス性能上のボトルネックを回避しながら、要求される高可用性の維持を可能としている。分散ストレージシステムの複数のデータノード上で複数種類のデータ構造への更新、変換を同時に行うことは、性能上、ボトルネックとなりやすい。本発明の好ましい態様においては、Ｗｒｉｔｅに特化した構造（Ｗｒｉｔｅの応答性能を優先した中間データ保持構造）を用意し、可用性保証のための複製実行時には、同期式（Ｓｙｎｃ）で中間構造で複製し、該中間構造で保持されるデータを非同期（Ａｓｙｎｃ）で正式のターゲット構造に変換する。

【００３１】

さらに、本発明の好ましい態様によれば、データノードやデータ構造、非同期に構造変換を実行するための契機（トリガー）を制御可能にすることで、様々なアプリケーションや負荷変動に対応可能としている。

【００３２】

本発明の好ましい態様によれば、特に制限されるものではないが、例えば、テーブル単位で、データ配置、データ構造、パーティショニング（分割）をコントロール可能としている。

【００３３】

データ構造として、例えば、

- ロウストア（Ｒｏｗ－ｓｔｏｒｅ）：
  - ・追記型（データの格納領域に記録を追加）、
  - ・更新型、
- カラムストア（Ｃｏｌｕｍｎ－ｓｔｏｒｅ）：
  - ・圧縮の有無、
- ライトログ（例えばライト性能を優先するために更新情報を追記するための構造）：
- インデックス（検索用の索引データ）の有無：
- データの格納順をソート（Ｓｏｒｔｉｎｇ）しているか：
- 分割（Ｐａｒｔｉｔｉｏｎｉｎｇ）の有／無、分割数：
- 分割（Ｐａｒｔｉｔｉｏｎｉｎｇ）単位、アルゴリズム：

等の項目について組み合わせが選択される。

【００３４】

本発明の好ましい態様によれば、例えばデータをどのデータノードに置くかが制御の対象となるほか、どのデータ構造とするかも制御対象となる。

【００３５】

・例えばＷｒｉｔｅ要求のみが行われる場合、ライトログ（ＷｒｉｔｅＬｏｇ）や、追記型ロウテーブル（Ｒｏｗ－ｔａｂｌｅ）とする。

【００３６】

・あるいは、ＲｅａｄとＷｒｉｔｅの組み合せに対して、ロウテーブル（Ｒｏｗ－ｔａ

ｂｌｅ）が選択される。

【００３７】

・さらに、分析アプリケーションに対して、例えばカラムストア（あるいはカラム指向データベース）を選択する。カラムストア方式は、クエリ（Ｑｕｅｒｙ）に対してストレージのリードアクセスを効率化する。

【００３８】

・あるいは、分散処理に対して、パーティショニング（データ分割）の粒度を相対的に小さくし、集中処理に対してパーティショニングを大きくするか、パーティショニングを止める等の制御を行うようにしてもよい。

【００３９】

・さらに、中間構造で保持されるデータを非同期（Ａｓｙｎｃ）でターゲット構造へ変換するためのトリガー（契機）を制御対象としてもよい。

【００４０】

・あるいは、分析アプリケーションが必要とするデータ鮮度（データの新しさの尺度）によって、データ変換の契機を調整するようにしてもよい。

【００４１】

本発明の好ましい態様によれば、それぞれがデータ格納部（図１の１２）を備え、ネットワーク結合される複数のデータ配置ノード（データノード）を備えた分散ストレージシステムにおいて、クライアントからのデータ更新要求時等に行われる複製において、複製先の１つ又は複数のデータノードでは、更新要求を受けたデータベースにおけるデータ構造とは、異なる１つ又は複数種のデータ構造で複製データを前記データ格納部（図１の１２）に格納する。その際、前記データノードは、複製データを一旦中間構造を保持して更新要求に対する応答をクライアントに返し、前記更新要求とは、非同期で目的のデータ構造に変換して格納する。

【００４２】

本発明の態様の１つによれば、データ構造情報（例えばデータ構造の管理情報やデータ配置特定情報）を保持管理する装置（図１の９）を備え、データアクセス手段（図１の６１１）およびデータノードをアクセスする手段（図４の１１２）は、データ構造情報を用いて、複製対象のデータに対するデータ構造（物理構造）を決定する。このため、分散ストレージノード毎に、複製データを異なるデータ構造で保持することができる。

【００４３】

本発明の態様の１つによれば、分散ストレージシステムにおいて、複製先のデータノードは、クライアントからの更新要求に対して、更新処理性能を優先する中間構造（中間データ保持構造、中間バッファ構造ともいう）に、データを一旦保持して、該更新要求に対して応答し、データ構造管理情報で指定されるデータ構造への変換を非同期で実行する。このため、複数種のデータ構造をそれぞれ中間データ保持構造に保持しつつ、更新処理の応答性能を維持できる。

【００４４】

本発明の態様の１つによれば、複数種のデータ構造を持ち、クライアント側が、アクセス内容に応じて適切なデータ構造に処理の振分け（適切なデータ構造を保持するデータノードをアクセスするように振り分ける）を行うようにしてもよい。このため、アクセス処理性能を向上することができる。

【００４５】

前記した関連技術を、上記本発明の態様の観点から分析する。

【００４６】

前述したとおり、関連技術の分散ストレージシステムにおいては、可用性保持のため、複製データを複数のノードで同一の物理構造で保持している。このため、可用性保持のための複製データの保持格納形式を制御することができない。

【００４７】

例えば、

・データの配置場所、
・データ配置（内部）構造、
・データを分散して格納するか、集中的に格納するかという格納方式、
等の複製データの保持格納形式について、可変に制御することができない。

【００４８】

データ移行等において、データ移行元のストレージ／データベースと、移行先のストレージ／データベースとは、同一データを異なるデータ構造で表現したものである、ということができる。例えば複製データを同一のデータ構造（物理構造）にて複数ノードで保持する構成において、互いに異なるデータ構造の各々について、各ノードで複製を保持する場合、ストレージ容量が過剰に必要とされる（この場合の複製に必要なストレージ容量は、データ容量×複製数×データ構造の種類の数）。そのため、計算機やディスク等のハードウェアを多く用意して利用することによって、購入コストや消費電力等の運用コストが増大してする（大量のデータコピー、大量のデータ構造の変換処理が必要とされる）。

【００４９】

また、関連技術において、分散ストレージシステムを利用するユーザ（例えばアプリケーション開発者）が、実現したいアプリケーション・ロジックを踏まえた上で、
・適切なデータ構造の選択、
・適切なスキーマの設計、
・適切なデータベースソフトウェア、設定の使い分け
を行う必要がある、ということである。いずれも、データベースシステムおよびストレージシステムに対して高い知見がユーザに要求されることから、これらをユーザ側で行うことは、実際上、困難である。

【００５０】

また、複製にあたり、適切なデータ構造を選択した場合であっても、複数のデータベースシステムを用意し、データの移行を行う必要がある、ということである。これらの処理は、計算機（サーバ）等において、データの入出力等の負荷が大きい。このため、移行先のデータベースのデータは、移行元のデータベースより古いデータとならざるを得ない。また、前述したように、同一内容のデータを互いに異なる複数のデータ構造として保持する場合、ストレージ利用効率が悪化してしまう。

【００５１】

本発明の態様の１つによれば、複製データを複数種のデータ構造（物理構造）で保持することで、要求される高可用性と、高速応答等性能を確保しつつ、データ構造変換のボトルネックを解消し、ストレージの利用効率を高めることが出来る。

【００５２】

以下、添付図面を参照して、いくつかの例示的な実施形態について説明する。

【００５３】

＜実施形態１＞

本発明の第１の例示的な実施形態について図面を参照して説明する。図１は、本発明の第１の実施形態のシステム構成の一例を示す図である。データノード１～４、ネットワーク５、クライアントノード６、構造情報管理手段（構造情報管理装置）９を備える。

【００５４】

データノード１～４は、分散ストレージを構成するデータ格納ノードであり、１つ以上の任意の数によって構成される。ネットワーク５は、データノード１～４を含むネットワークノード間の通信を実現する。クライアントノード６は、分散ストレージにアクセスする計算機ノードである。クライアントノード６は必ずしも独立して存在しなくてもよい。なお、データノード１～４がクライアント計算機を兼ねる例は、図２を参照して後述される。

【００５５】

データノード１～４は、それぞれ、データ管理・処理手段（データ管理・処理部）１１、２１、３１、４１、データ格納部１２、２２、３２、４２を備える。

【００５６】
データ管理・処理手段Ｘ１（Ｘ＝１、２、３、４）は、分散ストレージに対するアクセス要求を受け付け、処理を実行する。
【００５７】
データ格納部Ｘ２（Ｘ＝１、２、３、４）はデータノードの担当するデータの保持、記録を行う。
【００５８】
クライアントノード６は、クライアント機能実現手段（クライアント機能実現部）６１を備える。
【００５９】
クライアント機能実現手段６１は、データノード１～４によって構成される分散ストレージにアクセスする。
【００６０】
クライアント機能実現手段６１はデータアクセス手段（データアクセス部）６１１を備える。
【００６１】
データアクセス手段（データアクセス部）６１１は、構造情報管理手段９から構造情報（データ構造管理情報とデータ配置特定情報）を取得し、その構造情報を用いて、アクセス先のデータノードを特定する。
【００６２】
なお、各データノード１～４やネットワーク５内の任意の装置（スイッチ、中間ノード）において、構造情報管理手段９の構造情報保持部９２に格納される構造情報の一部又は全てを自装置内又は他の装置内のキャッシュ（不図示）に保持するようにしてもよい。
【００６３】
すなわち、以下の実施形態の動作の説明において、構造情報保持部９２に格納される構造情報に対するアクセスは、自装置内又は予め定められた所定の場所に配設されたキャッシュに対してアクセスするようにしてもよい。キャッシュに格納された構造情報の同期については、公知の分散システムの技術が適用できるため、ここでは詳細は省略する。よく知られているように、キャッシュを利用することでストレージ性能を高速化することが出来る。
【００６４】
構造情報管理手段（構造情報管理装置）９は、構造情報を変更する構造情報変更手段９１と、構造情報を保持する構造情報保持部９２を備える。構造情報保持部９２は、データ構造管理情報９２１（図４参照）とデータ配置特定情報９２２を含む（図４参照）。データ構造管理情報９２１は、後に図５を参照して説明されるが、テーブル識別子に対して、複製を特定するレプリカ識別子と、前記レプリカ識別子に対応したデータ構造の種類を特定するデータ構造情報と、指定されたデータ構造として格納されるまでの時間情報である更新契機からなるエントリをデータの複製数分有する。データ配置特定情報９２２は、後に図８を参照して説明されるが、テーブル識別子に対応して、前記レプリカ識別子と、前記レプリカ識別子に対応した１つ又は複数のデータ配置先のデータノード情報を有する。
【００６５】
本実施形態において、クライアントノード６は、データノード１～４とは独立に（別々に）設けることは必ずしも必要とされない。つまり、以下、変形例として説明するように、データノード１～４のうち、任意の１つ以上のノードに、クライアント機能実現手段６１を備えた構成としてもよい。
【００６６】
＜実施形態１の変形例＞
図２は、本発明の第１の実施形態の変形例の構成を示す図である。図２に示す通り、データノード１、２、３、４の各々に、クライアント機能実現手段６１が配設されている。
【００６７】

図２を参照すると、データノード１、２、３、４に配設されるクライアント機能実現手段６１は、図１のデータアクセス手段６１１の他に、構造情報キャッシュ保持部６１２を備える。

【００６８】

構造情報キャッシュ保持部６１２は、構造情報保持部９２に格納される構造情報の一部又は全てを格納するキャッシュメモリである。

【００６９】

構造情報同期手段（構造情報同期装置）９３は、構造情報のキャッシュの同期を制御する。構造情報保持部９２のデータを取得し、データノードのクライアント機能実現手段６１の構造情報キャッシュ保持部６１２の情報を更新する。

【００７０】

構造情報同期手段９３は、システムを構成する任意の機器に、任意の数、具備するようにしてもよい。例えば、各データノード１～４の少なくとも１つを実現する計算機上でソフトウェアとして動作させるようにしてもよい。

【００７１】

図２において、データノード１～４をそれぞれ個別の計算機として実現した場合の例を図３に示す。図３の例では、１つ以上の任意の数のデータノード計算機１０１～１０４と、ネットワーク１０５から構成される。

【００７２】

データノード計算機１０１～１０４は、それぞれＣＰＵ１０１ａ、データ記憶装置１０１ｂ、データ転送装置１０１ｃを備える。ＣＰＵ１０１ａにより、データ管理・処理手段１１、クライアント機能実現手段６１の機能の全て又は一部を実現する。

【００７３】

データ記憶装置１０１ｂは、例えば、ハードディスクドライブ、フラッシュメモリ、ＤＲＡＭ（Ｄｙｎａｍｉｃ Ｒａｎｄｏｍ Ａｃｃｅｓｓ Ｍｅｍｏｒｙ）、ＭＲＡＭ（Ｍａｇｎｅｔｏｒｅｓｉｓｔｉｖｅ Ｒａｎｄｏｍ Ａｃｃｅｓｓ Ｍｅｍｏｒｙ）、ＦｅＲＡＭ（Ｆｅｒｒｏｅｌｅｃｔｒｉｃ Ｒａｎｄｏｍ Ａｃｃｅｓｓ Ｍｅｍｏｒｙ）、ＰＲＡＭ（Ｐｈａｓｅ ｃｈａｎｇｅ ＲＡＭ）、ＲＡＩＤコントローラに結合された記憶装置、磁気テープのようにデータを記録可能な物理媒体、又は、ストレージノードの外部に設置された媒体にデータを記録する制御装置である。ネットワーク１０５及びデータ転送装置１０１ｃは、例えばＥｔｈｅｒｎｅｔ（登録商標）、Ｆｉｂｒｅ ＣｈａｎｎｅｌやＦＣｏＥ（Ｆｉｂｒｅ Ｃｈａｎｎｅｌ ｏｖｅｒ Ｅｔｈｅｒｎｅｔ（登録商標））、ＩｎｆｉｎｉＢａｎｄ（Ｉｎｔｅｌ社その他による団体が推進する高速ＩＯバスアーキテクチャ）、ＱｓＮｅｔ（Ｑｕａｄｒｉｃｓ社製品）、Ｍｙｒｉｎｅｔ（Ｍｙｒｉｃｏｍ社製品）、Ｅｔｈｅｒｎｅｔ（登録商標）、あるいはこれらを利用するＴＣＰ／ＩＰ（Ｔｒａｎｓｍｉｓｓｉｏｎ／Ｃｏｎｔｒｏｌ Ｐｒｏｔｏｃｏｌ／Ｉｎｔｅｒｎｅｔ Ｐｒｏｔｏｃｏｌ）やＲＤＭＡ（Ｒｅｍｏｔｅ Ｄｉｒｅｃｔ Ｍｅｍｏｒｙ Ａｃｃｅｓｓ）のような上位プロトコルによって実現しうる。ただし、ネットワーク１０５の実現方法は、これらに限られない。Ｅｔｈｅｒｎｅｔ（登録商標）で実現する場合の例としては、データ転送装置１０１ｃは計算機に接続されるネットワークカード、ネットワーク１０５はＥｔｈｅｒｎｅｔ（登録商標）ケーブルおよびスイッチ等から構成される。

【００７４】

データノード１～４の実現は、仮想化された計算機（Ｖｉｒｔｕａｌ Ｍａｃｈｉｎｅ）であってもよい。代表的な例としてＶＭＷａｒｅ（ＶＭＷａｒｅ社製品）、Ｘｅｎ（Ｃｉｔｒｉｘ社商標）等がある。

【００７５】

＜データノードの詳細の一例＞

図４は、本発明の第１の実施形態の構成例をより詳細に説明する図である。図４には、図１のデータノード１～４を中心に示した構成が示されている。なお、図４等の図面において、簡単化のため、構造情報保持部９２に格納される構造情報は参照符号９２で参照さ

れる場合がある。

【００７６】

データノードのデータ管理・処理手段１１は、アクセス受付手段（アクセス受付部）１１１、アクセス処理手段（アクセス処理部）１１２、データ構造変換手段（データ構造変換部）１１３を備えている。

【００７７】

アクセス受付手段１１１は、データアクセス手段６１１からアクセス要求を受け付け、処理完了後にデータアクセス手段６１１に応答を返す。

【００７８】

アクセス処理手段１１２は、構造情報保持部９２の構造情報（あるいはその任意の場所に保持されるキャッシュ情報）を用い、アクセス処理を、該当するデータ格納部１２Ｘ（Ｘ＝１、２、３）に対して処理を行う。

【００７９】

データ構造変換手段１１３は、一定契機毎に構造別データ格納部１２１のデータを用いて、構造別データ格納部１２Ｘ（Ｘ＝１、２、３）に変換する。

【００８０】

データ格納部１２は、複数種の構造別データ格納部を備えている。図４では、構造別データ格納部１２１（データ構造Ａ）、構造別データ格納部１２２（データ構造Ｂ）、構造別データ格納部１２３（データ構造Ｃ）を備える。

【００８１】

どのようなデータ構造を選択するかは、構造別データ格納部１２Ｘ（X＝１、２、３）単位で任意である。

【００８２】

本実施形態では、構造別データ格納部１２１（例えばデータ構造Ａ）は、データの書き込みを伴う処理（データの追加や更新）に対する応答性能に特化した構造をとる。具体的には、データ変更内容をキュー（例えばＦＩＦＯ（Ｆｉｒｓｔ　Ｉｎ　Ｆｉｒｓｔ　Ｏｕｔ））として高速なメモリ（デュアルポートＲＡＭ等）上に保持するソフトウェア、アクセス要求処理内容を任意の記憶媒体にログとして追記するソフトウェア等が実装される。データ構造Ｂ、データ構造Ｃは、データ構造Ａとは異なるデータ構造であり、互いに異なるデータアクセス特性を持つ。

【００８３】

データ格納部１２は、必ずしも単一の記憶媒体でなくてもよい。図４のデータ格納部１２を複数のデータ配置ノードからなる分散ストレージシステムとして実現し、各構造別データ格納部１２Ｘを分散して格納する方式であってもよい。

【００８４】

データ配置特定情報９２２は、分散ストレージに格納するデータ、あるいはデータ断片の格納先を特定するための情報（および情報を格納、取得する手段）である。データの分散配置方式は、前述した通り、例えばメタサーバ方式や分散ＫＶＳ方式が一般的に利用される。

【００８５】

メタサーバ方式の場合、データの位置情報を管理する情報（例えばブロックアドレスとその対応するデータノードアドレス）がデータ配置特定情報９２２である。メタサーバは、この情報（メタデータ）を参照することで、必要なデータの配置先を知ることが出来る。

【００８６】

分散ＫＶＳ方式の場合、システムに参加するノードのリストが、このデータ配置特定情報に該当する。データを格納する識別子と、ノードリスト情報を用いることによって、データ格納先のデータノードを決定することが出来る。

【００８７】

データアクセス手段６１１は、構造情報管理手段９におけるデータ配置特定情報９２２

、あるいは、予め定められた所定の場所に記憶されるデータ配置特定情報９２２のキャッシュ情報を用いて、アクセスすべきデータノード１～４を特定し、データノードのアクセス受付手段１１１に対してアクセス要求を発行する。

【００８８】

＜データ構造管理情報＞

データ構造管理情報９２１は、データの集合毎にデータの格納方式を特定するためのパラメータ情報である。図５は、図４のデータ構造管理情報９２１の一例を示す図である。特に制限されるものではないが、本実施形態では、データの格納方式を制御する単位を、テーブルとする。そして、テーブル毎（テーブル識別子毎）に、レプリカ識別子、データ構造の種別、更新契機の各情報を、データ複製の複製数分、用意する。

【００８９】

図５（Ａ）では、各テーブルは、可用性確保（保持）のために、３つの複製を保持する。レプリカ識別子は、それぞれの複製を特定する情報であり、図５（Ａ）では、０、１、２として付与されている。

【００９０】

データ構造は、データの格納方式を示す情報である。図５（Ａ）では、３種類のデータ構造（Ａ、Ｂ、Ｃ）をレプリカ識別子毎に異なる方式を指定している。

【００９１】

図５（Ｂ）に、データ構造Ａ、Ｂ、Ｃの例を示す。データの格納方式の種類として、
Ａ：キュー、
Ｂ：ロウストア、
Ｃ：カラムストア
が指定されている。

【００９２】

この場合、テーブル識別子「Ｓｔｏｃｋｓ」のレプリカ識別子０は、データ構造Ｂ（ロウストア）として格納される。

【００９３】

データ構造は、それぞれデータを格納するための方式であり、
Ａ：キュー（ＱＵＥＵＥ）は、リンクトリスト（Ｌｉｎｋｅｄ　Ｌｉｓｔ）である。

【００９４】

Ｂ：ロウストア（ＲＯＷ　ＳＴＯＲＥ）は、テーブルのレコードを行（ＲＯＷ）順に格納する。

【００９５】

Ｃ：カラムストア（ＣＯＬＵＭＮ　ＳＴＯＲＥ）は、列（ＣＯＬＵＭＮ）順に格納する。

【００９６】

図６に、テーブルのデータ保持構造の一例を示す。図６の（Ａ）のテーブルは、Ｋｅｙカラムと、３つのＶａｌｕｅカラムを備え、各ローは、Ｋｅｙと３つのＶａｌｕｅのセットからなる。

【００９７】

カラムストア、ロウストアは、それぞれ図６に示すように、記憶媒体上の格納順序を行（ロー）ベース、列（カラム）ベースに格納されている形式のことを指す。

【００９８】

図６では、テーブル（図６の（Ａ）参照）の格納方式として、
レプリカ識別子０と１のデータとして、データ構造Ｂ（ロウストア）で保持し（図６の（Ｂ）、（Ｃ）参照）、
レプリカ識別子２のデータとして、データ構造Ｃ（カラムストア）として保持する（図６の（Ｄ）参照）。

【００９９】

再び図５（Ａ）を参照すると、データ構造管理情報９２１（図４参照）における更新契

機は、データを指定されたデータ構造として格納されるまでの時間契機である。Ｓｔｏｃｋｓのレプリカ識別子０の例では３０ｓｅｃと指定されている。したがって、Ｓｔｏｃｋｓのレプリカ識別子０のデータ構造Ｂ（ロウストア）を格納するデータノードにおいて、ロウストア方式の構造別データ格納部１２２に対して、データの更新が反映されるのが３０ｓｅｃ契機であることを示す。データ更新が反映されるまでの間は、キュー等の中間構造としてデータが保持される。また、データノードでは、クライアントからの要求に対しても、中間構造に格納して応答が行われる。本実施形態では、指定されたデータ構造への変換は、更新要求とは、非同期（Ａｓｙｎｃｈｒｏｎｏｕｓ）で行われる。

【０１００】

図７は、テーブルのデータ保持、非同期更新の例を模式的に説明する図である。更新契機が「０」より大きい場合には、各データノードは、Ｗｒｉｔｅ（更新要求）の応答速度に優れた構造を中間構造として持ち、更新内容を受け付ける。中間構造に書き込みを行った時点で、更新要求元のクライアントに対して処理完了の応答を返す。

【０１０１】

各データノードの中間構造（Ｗｒｉｔｅ向け中間構造、Ｗｒｉｔｅ優先中間構造、あるいは「中間データ保持構造」ともいう）に書き込まれた更新データは、各データノードにおいて、それぞれ、データ構造Ｂ、Ｃに、それぞれ非同期（Ａｓｙｎｃ）に更新される。図７に示す例では、Ｗｒｉｔｅにより、レプリカ識別子が０のデータノードにおいて、Ｗｒｉｔｅ向け中間構造には、データ構造Ａが格納保持され、レプリカ識別子１、２のデータノードに対して同期方式（Ｓｙｎｃｈｒｏｎｏｕｓ）で、Ｗｒｉｔｅ向け中間構造に保持されたデータ構造Ａのデータがレプリケート（複製）され、レプリカ識別子１、２のデータノードの各々において、Ｗｒｉｔｅ向け中間構造にはデータ構造Ａのデータが一旦格納保持される。レプリカ識別子０、１、２に対応するデータ構造にそれぞれ対応するデータノードにおいて、ターゲットのデータ構造Ｂ、Ｂ、Ｃへの変換は、図５（Ａ）に示すようなデータ構造管理情報９２１の更新契機情報により指定される。

【０１０２】

図７に示すように、一つのデータノードのＷｒｉｔｅ向け中間構造に書き込まれた更新データ（データ構造Ａ）のデータノード間での複製は、書き込み（更新）と同期（Ｓｙｎｃ）して行われる。このような構成をとることによって、Ｗｒｉｔｅ（書き込み）データに対して、すぐにＲＥＡＤ（読み出し）系のアクセスがないデータに対しては、Ｗｒｉｔｅの応答速度を高めることが出来る。

【０１０３】

また、（後の）ＲＥＡＤ系アクセス時には、当該ＲＥＡＤアクセスに必要なデータ構造に既に変換されているため、変換されたデータ構造を用いて、ＲＥＡＤ系アクセスを処理することで、処理の高速化を実現することができる。さらに、ＲＥＡＤ系アクセスの種類によって、適切なデータ構造を選んでアクセス先ノードを使い分けることも出来る。

【０１０４】

本実施形態では、単に説明の簡易化のため、データ構造の種類の数をＡ、Ｂ、Ｃの３つとしたが、データ構造の種類の数は３つに制限されるものでないことは勿論であり、特性の異なる任意の複数種類であってもよい。また、データ構造の例として、キュー、カラムストア、ロウストアの３種を例示したが、かかる例に制限されるものでないことは勿論である。例えば、

・ロウストア構造におけるインデックスの有無、
・インデックスを作成したカラムの種類の違い、
・更新を追記構造で格納するロウストア形式、
等であってもよい。

【０１０５】

図５に示した例とは異なる方式として、データ構造管理情報９２１において、データ構造の種類の代わりに、データ格納プログラムを指定するようにしても良い。例えば、図５（Ａ）のデータ構造Ａとしてデータをキューに格納するプログラムＡ、データ構造Ｂ、Ｃ

として異なるデータベース・ソフトウェアを指定する。この場合、データ構造Ａが指定されているテーブルのレプリカ識別子を格納するデータノードでは、受け付けたデータをプログラムＡを実行することで処理する。

【０１０６】

＜データ配置特定情報＞

図８は、図４のデータ配置特定情報９２２の例を示す。各テーブル識別子のレプリカ識別子０、１、２毎に、配置ノードが指定されている。これは、前述したメタサーバ方式に対応している。

【０１０７】

＜分散ＫＶＳ＞

分散ＫＶＳ方式の場合、データ配置特定情報９２２は、分散ストレージに参加しているノードリスト情報（不図示）が該当する。このノードリスト情報をデータノード間で共有することによって、「テーブル識別子」＋「レプリカ識別子」をキー情報として、コンシステント・ハッシング方式により、配置ノードを特定することが出来る。また、レプリカの配置先として、コンシステント・ハッシング方式における隣接ノードに格納することができる。コンシステント・ハッシング方式は第４の実施形態で説明する。

【０１０８】

再び図８を参照すると、データ配置特定情報９２２において、配置ノードは、可用性を保証するためには、同一のテーブルが同一ノードに保持されることがないように指定されなければならない。

【０１０９】

例えば、図５（Ａ）のＳｔｏｃｋｓテーブルのレプリカ識別子０と１と２の配置ノードは互いに重複してはならない。なお、可用性の考慮を無視するのであれば、この制限はこの限りではない。つまり、複数種類のレプリカを同一ノードに保持してもよい。

【０１１０】

＜Ｗｒｉｔｅ処理のシーケンス＞

本発明の第１の実施形態の動作について説明する。図９は、図１乃至図８を参照して説明した本発明の第１の実施形態におけるＷｒｉｔｅ処理（更新を伴う処理）のシーケンスを示す図である。

【０１１１】

クライアント機能実現手段６１は、構造情報管理手段９の構造情報保持部９２に保持されているデータ配置特定情報９２２（図４、図８参照）の情報を取得する（あるいは任意場所のキャッシュメモリから情報を取得する）。

【０１１２】

クライアント機能実現手段６１は、取得した情報を用いて、Ｗｒｉｔｅ処理を行うデータの配置先のデータノード（図９では、レプリカ識別子０のデータノード１）に対して、Ｗｒｉｔｅアクセス命令を発行する。

【０１１３】

データノード１のアクセス受付手段１１１は、Ｗｒｉｔｅアクセス要求（Ｗｒｉｔｅ処理要求）を受け付け、レプリカ識別子１、２に指定されているデータノード２、３に対してＷｒｉｔｅアクセスを転送する。レプリカ識別子１、２のデータノードを特定する方法としては、データノード１が構造情報保持部９２（あるいは適切なキャッシュ）にアクセスしても良いし、クライアント実現手段６１が発行するＷｒｉｔｅアクセス命令にデータ構造管理情報９２１の全部あるいは一部の情報をともに渡すようにしてもよい。

【０１１４】

各データノードのアクセス処理手段１１２は、受け取ったＷｒｉｔｅアクセス要求の処理を行う。

【０１１５】

アクセス処理手段１１２は、データ構造管理情報９２１の情報を参照して、Ｗｒｉｔｅ処理を実行する。

【０１１６】
更新契機が「０」より大きい場合には、Ｗｒｉｔｅ処理内容をデータ構造Ａの構造別データ格納部１２１に格納する。
【０１１７】
更新契機が「０」の場合には、データ構造管理情報９２１に指定されているデータ構造の構造別データ格納部１２Ｘに対して格納する。
【０１１８】
アクセス処理手段１１２は、Ｗｒｉｔｅ処理完了後、アクセス受付手段１１１に、完了通知を発行する。
【０１１９】
レプリカ先のデータノード（２、３）は、レプリカ元のデータノード１のアクセス受付手段１１１にＷｒｉｔｅ完了応答を返答する。
【０１２０】
アクセス受付手段１１１は、データノード１のアクセス処理手段１１２からの完了通知と、各レプリカ先のデータノード２、３の完了通知を待ち合わせ、全て受け取った後に、クライアント機能実現手段６１に対して応答する。
【０１２１】
データ構造変換手段１１３（図４参照）は、定期的に構造別データ格納部１２１（データ構造Ａ）のデータを、構造別データ格納部１２Ｘ（データ構造管理情報９２１に指定されている、最終格納先データ構造）に変換して格納する。
【０１２２】
なお、図９の例では、データノード１が、レプリカ先のデータノード２、３に対して、Ｗｒｉｔｅアクセスを転送しているが、図１０に示すように、クライアント機能実現手段６１が、格納先のデータノードの全てに対して、Ｗｒｉｔｅアクセスを発行するようにしても良い。
【０１２３】
図１０の例では、図９と比較して、Ｗｒｉｔｅアクセス要求の待ち合わせをクライアント機能実現手段６１が行うことが異なる。
【０１２４】
＜参照系処理のシーケンス＞
図１１は、本発明の第１の実施形態における参照系処理（ＲＥＡＤ処理）のシーケンスを示す図である。
【０１２５】
クライアント計算機（クライアントノード）６は、データ構造管理情報９２１の情報を取得して、命令の実行先ノードを特定する。レプリカデータを配置するノードは、レプリカ識別子のいずれを用いてもよいが、行う処理によって適切なノードを選択することが望ましい。
【０１２６】
参照系処理とは、データの読み込みを伴う処理をいい、例えばＳＱＬ（Ｓｔｒｕｃｔｕｒｅｄ　Ｑｕｅｒｙ　Ｌａｎｇｕａｇｅ）文におけるＳｅｌｅｃｔ文による命令等に対応する。
【０１２７】
また、
　あるテーブルＡからデータを読み出し、
　当該データを用いた演算結果をテーブルＢに更新する場合、
テーブルＡからのデータ読み出しは参照系処理に該当する。
【０１２８】
あるいは、テーブルＡを参照した後、テーブルＡを更新するような処理の場合、一括してＷｒｉｔｅ処理（図９、図１０記載）として扱っても良い。あるいは、テーブルＡの参照処理は参照系処理として扱い、テーブルＡの更新を、更新処理として扱ってもよい。

【０１２９】
＜クライアント機能実現手段の動作＞
図１２は、クライアント機能実現手段６１の視点によるアクセス処理の動作を説明するフローチャートである。図１２を参照して、クライアントのアクセスフローについて説明する。
【０１３０】
まず、クライアント機能実現手段６１が、構造情報保持部９２の情報をマスタ、あるいは任意の箇所のキャッシュにアクセスすることで取得する（図１２のステップＳ１０１）。
【０１３１】
次に、クライアントが発行する命令内容がＷｒｉｔｅ処理であるか参照処理（Ｒｅａｄ）であるかを識別する（ステップＳ１０２）。
【０１３２】
これは、発行命令のコマンドにより指定したり、命令の実行コードを解析したりすることで特定することが出来る。例えば、ＳＱＬを処理するストレージシステムの場合、
・ＩＮＳＥＲＴ命令（テーブルへレコードを追加するＳＱＬ命令）であれば、Ｗｒｉｔｅ処理、
・ＳＥＬＥＣＴ命令（テーブルからレコードを削除するＳＱＬ命令）であれば、参照系処理、
である。
【０１３３】
あるいは、クライアント機能実現手段６１を用いて、命令を呼び出す際に、明示的に指定するようにしても良い（そのようなＡＰＩ（Ａｐｐｌｉｃａｔｉｏｎ Ｐｒｏｇｒａｍ Ｉｎｔｅｒｆａｃｅ）を準備する）。
【０１３４】
ステップＳ１０２の結果、Ｗｒｉｔｅ処理であれば、ステップＳ１０３以降に進む。
【０１３５】
Ｗｒｉｔｅ処理の場合、クライアント機能実現手段６１は、更新が必要なノードをデータ配置特定情報９２２の情報を用いて特定する。この処理は、図９を参照して説明した通りである。
【０１３６】
クライアント機能実現手段６１は、特定したデータノードに対して、命令実行要求（更新要求）を発行する（ステップＳ１０３）。
【０１３７】
クライアント機能実現手段６１は、更新要求発行先のデータノードからの応答通知を待ち合わせ、更新要求が、各データノードに保持されたことを確認する（ステップＳ１０４）。
【０１３８】
図１２は、クライアント機能実現手段６１が、更新先のデータノードに対して命令を発行し、応答を待ち合わせるという図１０のシーケンスに対応するクライアント機能実現手段６１の動作を説明するためのフローチャートである。
【０１３９】
ステップＳ１０２の結果、参照処理である場合には、ステップＳ１０５へ進む。
【０１４０】
ステップＳ１０５では、まず、クライアント機能実現手段６１は、処理内容の特性を特定（認識）する（ステップＳ１０５）。
【０１４１】
クライアント機能実現手段６１は、特定した処理特性と、その他のシステム状況を踏まえて、アクセス対象のデータノードを選択し、命令要求を発行する処理を行う（ステップＳ１０６）。

【０１４２】
クライアント機能実現手段６１は、その後、データノードからアクセス処理結果を受け取る（ステップＳ１０７）。
【０１４３】
以下、ステップＳ１０５、ステップＳ１０６の処理について説明を補充する。
【０１４４】
まず、クライアント機能実現手段６１は、データ構造管理情報９２１に格納されている情報から、アクセス対象のデータが保持されているデータ構造の種類を知ることが出来る。例えば、図５（Ａ）の例の場合、ＷＯＲＫＥＲＳテーブルにアクセスする場合、レプリカ識別子０、１は、データ構造Ｂ、レプリカ識別子２は、データ構造Ｃである。
【０１４５】
そして、クライアント機能実現手段６１では、データノードに対して行われるデータアクセスが、どちらのデータ構造に適しているかを判断し、適している方を選択する。
【０１４６】
より詳しくは、例えば、クライアント機能実現手段６１では、アクセス要求であるＳＱＬ文を解析し、テーブル識別子が「ＷＯＲＫＥＲＳ」のテーブル内のあるカラムの総和をとる命令の場合には、データ構造Ｃ（カラムストア）を選択し、ある特定のレコードを取り出す命令の場合には、データ構造Ｂ（ロウストア）が向いていると判断する。
【０１４７】
ある特定のレコードを取り出す命令であった場合、レプリカ識別子０、１では、どちらを選択しても良い。なお、必ずしも「最新のデータで処理を行う必要が無い場合」、レプリカ識別子１（更新契機は３０ｓｅｃ）を用いることが望ましい。
【０１４８】
この「最新のデータで処理を行う必要が無い場合」であることの特定は、アプリケーション・コンテキストに依存する。このため、クライアント機能実現手段６１に受け渡される命令に、利用するデータ構造や、必要なデータの鮮度（データの新しさ）を特定する情報を、明示的に指定する形式としても良い。
【０１４９】
アクセスすべきレプリカ識別子（データ構造）を特定した後、アクセスすべきデータノードを算出する。このとき、分散ストレージシステムの状況に応じて、アクセスノードの選択を変更できるようにしても良い。例えば、あるテーブルが同一のデータ構造Ｂとして、データノード１、２に格納されている際に、データノード１のアクセス負荷が大きい場合に、データノード２を選択するような動作に変更してもよい。
【０１５０】
また、別のデータ構造Ｃとして、データノード３に格納されている場合に、データノード３のアクセス負荷が、データノード１、２と比較して小さければ、処理するアクセス内容がデータ構造Ｂの方が向いていたとしても、データノード３（データ構造Ｃ）に対して、アクセス要求を発行するようにしても良い。
【０１５１】
クライアント機能実現手段６１では、このようにして算出・選択されたデータノードに対して、アクセス要求を発行し（ステップＳ１０６）、該データノードから、アクセス処理結果を受け取る（ステップＳ１０７）。
【０１５２】
＜データノードの動作＞
図１３は、図４のデータノードにおけるアクセス処理を説明するフローチャートである。図１３、図４を参照して、データノードの動作について詳細に説明する。
【０１５３】
まず、データノードのデータ管理・処理手段１１のアクセス受付手段１１１がアクセス処理要求を受け付ける（図１３のステップＳ２０１）。
【０１５４】

次に、データノードのデータ管理・処理手段１１のアクセス受付手段１１１は、受け付けた処理要求の内容がＷｒｉｔｅ処理であるか、参照処理であるか判定する（ステップＳ２０２）。

【０１５５】

ステップＳ２０２の結果、Ｗｒｉｔｅ処理であった場合、データノードのデータ管理・処理手段１１のアクセス処理手段１１２は、構造情報保持部９２におけるデータ構造管理情報９２１の情報を取得する（ステップＳ２０３）。データ構造管理情報９２１の情報取得は、マスタデータにアクセスしてもよいし、任意の箇所にあるキャッシュデータにアクセスするようにしてもよいし、あるいは、図１又は図２のクライアント機能実現手段６１が、データノードに対して発行する要求に情報（マスタデータ又はキャッシュデータへのアクセス）を付与し、アクセス処理手段１１２では、その情報を用いてアクセスするようにしてもよい。

【０１５６】

次に、アクセス処理手段１１２は、データ構造管理情報９２１の情報から、該データノードに対する処理の更新契機が「０」（零）であるかどうかを判定する（ステップＳ２０４）。

【０１５７】

ステップＳ２０４の結果、更新契機が「０」の場合、アクセス処理手段１１２は、構造情報保持部９２の構造情報に指定されたデータ構造を、直接、更新する（ステップＳ２０５）。すなわち、更新データを指定されたデータ構造に変換し対応する構造別データ格納部１２Ｘ（Ｘ＝１、２、３）に格納する。

【０１５８】

更新契機が「０」でない場合、アクセス処理手段１１２は、Ｗｒｉｔｅ向け中間構造（構造別データ格納部１２１）に、更新データを格納する（ステップＳ２０６）。

【０１５９】

ステップＳ２０５、２０６の場合、いずれも、処理完了後、アクセス受付手段１１１は、要求元のクライアント実現手段６１に対して、処理完了通知を応答する（ステップＳ２０７）。

【０１６０】

ステップＳ２０２の結果、データの参照処理であった場合、参照処理の実行を行う（ステップＳ２０８）。

【０１６１】

参照処理の実行方式として、特に制限されるものでないが、代表的には、以下の３種類の方法を挙げることができる。

【０１６２】

（１）第１の方法は、データ構造管理情報９２１に指定されているデータ構造のデータ格納部のデータを利用して処理する。これは最も性能が優れるが、更新契機が大きい場合には、Ｗｒｉｔｅ向け中間構造のデータが参照処理に反映されていない可能性がある。このため、データの不整合が生じる可能性がある。ただし、アプリケーション開発者が事前に認識していて利用する場合や、Ｗｒｉｔｅ後に、データの読み出しが更新契機内に起きないことがわかっているか、もし新しいデータアクセスが必要な場合には、更新契機が「０」のレプリカ識別子データにアクセスすると決めている場合には、特に、問題はない。

【０１６３】

（２）第２の方法は、別途行われる変換処理の適用を待ってから処理する方法である。これは、実装が容易であるが、応答性能が劣化する。応答性能を求めないアプリケーションの場合、問題はない。

【０１６４】

（３）第３の方法は、データ構造管理情報９２１に指定されているデータ構造と、Ｗｒｉｔｅ向け中間構造に保持されているデータの両方を読んで処理する。この場合、常に、最新のデータを応答できるが、第１の方法より性能が劣化する。

【０１６５】
上記第１乃至第３のいずれの方法をとってもよい。また、複数の種類を実現し、システムの設定ファイルとして記述する、クライアント機能実現手段６１から発行される処理命令の中に、どの方法で実行するかを指定するようにしてもよい。
【０１６６】
＜データ構造変換手段の変換動作＞
図１４は、図４のデータ構造変換手段１１３におけるデータ変換処理の動作を示すフローチャートである。図１４、図４を参照して、データ変換処理を説明する。
【０１６７】
データ構造変換手段１１３は、定期的に変換処理の必要の有無を判定するため、データノード内のタイマ（図４では不図示）でのタイムアウト発生による呼び出しを待つ（図１４のステップＳ３０１）。なお、このタイマは、専用タイマとしてデータ構造変換手段１１３内に備えるようにしてもよい。タイマのタイムアウト時間は、図５（Ａ）の更新契機（sec）に対応する。
【０１６８】
次に、構造情報保持部９２の構造情報（データ情報）を取得し（ステップＳ３０２）、変換が必要なデータ構造があるか否かを判定する（ステップＳ３０３）。例えば、タイマで判定が１０秒毎に行われるときに、更新契機が２０秒のデータ構造は、２０秒毎に変換処理を実行するため、１０秒時点では、変換処理を行わなくても良い。
【０１６９】
変換処理が必要でない場合には、タイマ呼び出し待ち（タイマでのタイムアウト発生により呼び出されるまでウエイト）に戻る（ステップＳ３０１）。
【０１７０】
一方、変換処理が必要な際には、更新向け中間データ構造から、変換対象のデータに対する更新処理内容を読み出し（ステップＳ３０４）、変換先の構造別データ格納部１２Ｘ（Ｘ＝１～３）へ更新情報を反映する処理を行う（ステップＳ３０５）。
【０１７１】
＜実施形態２＞
本発明の第２の実施の形態について説明する。本発明の第２の実施の形態では、データを、所定単位で複数に分割して、複数のデータノードに格納できるようにしている。本実施形態のシステムの基本構成は、図１、図２、図４等に示した構成とされるが、図１５、図１６を参照して説明されるように、本実施形態においては、データ構造管理情報９２１、データ配置特定情報９２２の内容が拡張されている。また、図１７を参照して説明されるように、本実施形態においては、データノードのアクセス受付手段が、アクセス処理手段にアクセス要求を発行するときに、他のデータノードのアクセス処理手段に対しても、アクセス要求を発行し、さらに、データ構造変換手段が、他のデータノードのデータ構造変換手段に対して、変更要求を発行する構成とされていることが、前記第１の実施形態と相違している。なお、本実施形態におけるデータノードの構成も、基本的には、図４に従うが、その詳細は図１７を参照して後述される。
【０１７２】
本実施形態では、格納対象とするデータ（テーブル識別子）を、複製の格納単位（レプリカ識別子）毎に、パーティショニング（分割）して、分割した格納単位を、各データノードでそれぞれ格納することができる。
【０１７３】
図１５は、データ構造管理情報９２１（図４参照）の例を示す図である。データ構造管理情報９２１は、テーブル識別子に対して、複製数分、レプリカ識別子と、該レプリカ識別子に対応したパーティション数を備える。
【０１７４】
パーティション数が「１」であるレプリカ識別子は、複製（レプリカ）を１つのデータノードに格納する。その場合の動作は、前記第１の実施形態と同一である。

【０１７５】
パーティション数が「１」よりも大きい場合、そのレプリカ識別子のデータを、複数のデータノードに分割して格納する。図１６は、その場合のデータ配置特定情報９２２の例を示す図である。
【０１７６】
データ構造管理情報９２１において、あるレプリカ識別子のパーティション数が「１」よりも大きい場合、データ配置特定情報９２２（図４参照）において、当該レプリカ識別子に対して、図１６に示すように、配置ノードのリスト（分割して格納する複数のデータノードのリスト）を記録する。
【０１７７】
図１５のデータ構造管理情報９２１の例では、テーブル識別子「ＷＯＲＫＥＲＳ」のレプリカ識別子２のパーティション数が「４」である。図１６のデータ配置特定情報９２２では、テーブル識別子「ＷＯＲＫＥＲＳ」のレプリカ識別子２の「配置ノード」として、ノード番号２、３、５、６が指定されている。
【０１７８】
配置ノードの決定は、テーブル識別子毎に、システム全体として想定される要求可用性レベルを保つように決める。マニュアル（人手）で行ってもよいし、図１５のデータ構造管理情報９２１、図１６のデータ配置特定情報９２２の内容をプログラムで自動生成するようにしてもよい。
【０１７９】
例えば、一般的に、可用性レベルは、複製数（レプリカ数）に応じて決定される。求める可用性レベルが３レプリカであれば、レプリカ識別子を３つ用意し、それぞれの配置ノードが互いに重複しないように決定する。
【０１８０】
図１６の例では、テーブル識別子「ＷＯＲＫＥＲＳ」のレプリカ識別子の各配置ノードは、互いに重複しないよう指定されている。なお、レプリカ識別子を４つ以上用意してもよいことは勿論である。例えばレプリカ識別子が４つの場合、求める可用性レベルが「３」のままであれば、同一のテーブル識別子のレプリカ識別子の配置ノードとして、１つまで重複して選ぶことが出来る（例えば、４つのレプリカ識別子のうち、配置ノードが重複するレプリカ識別子が２つあってもよい）。
【０１８１】
各レプリカ識別子のデータ格納構造と、分割配置戦略（パーティショニング・ストラテジ）により、パーティションニング時の配置ノードの重複を許すか否かが異なる。
【０１８２】
例えば、次のような場合には、パーティションニング時の配置ノードを重複して格納することが出来る。ノード番号１－１８のデータノードに、ロウストア形式（データ構造Ｂ）で、１２分割のレプリカを、２つ格納する場合、互いに重複を許さない場合には格納が不可能である。しかし、この場合、次のようにすれば、２レプリカ・レベルの可用性を満たしつつ、配置ノードを重複させて割り当てることが出来る。
【０１８３】
レプリカ識別子０は、ノード番号１－１２、
レプリカ識別子１は、ノード番号７－１８、
に分割して格納するものとする。
【０１８４】
このとき、レプリカ識別子０と１の同一レコードのデータが、同一ノードに格納されないように、分割配置戦略が決定されていれば、可用性レベルを満たすことが出来る。具体的には、下記のように、テーブルをパーティショニングする際に、ある任意のカラムの値によって、分散配置する場合（カラムの値の前半、後半で分割）、
・レプリカ識別子０のノード番号１－６には、カラムの値の前半、ノード番号７－１２にはカラムの値の後半、

・レプリカ識別子１のノード番号７－１２には、カラムの値の前半、ノード番号１３－１８には、カラムの値の後半
というように格納することで、同一のレコードが、同一のノードに格納されることは回避される。このようにすることで、配置ノードの割り当てを重複させながら、可用性を満たすことが出来る。

【０１８５】
配置ノード先の決定は、システムあるいはテーブル識別子毎に指定される可用性レベルを満たすように行う。

【０１８６】
パーティション数が「１」よりも大きいレプリカ識別子に対する更新時のアクセス先は、配置ノード群のいずれを選んでも良い。あるいは、常に、リストの最初のノードを選ぶようにしてもよい（例えばテーブル識別子「ＷＯＲＫＥＲＳ」のレプリカ識別子「２」の場合、ノード番号２のデータノード）。後者の方が、データ構造変換手段１１３における、構造別データ格納部１２１から構造別データ格納部１２２、１２３への変換処理がやや簡略化される。

【０１８７】
パーティション時には、コンシステント・ハッシング法等を用いて分散配置してもよいし、前述したようなテーブルのあるカラムの値や、ユニークなＫｅｙの範囲などで格納先を決定してもよい。

【０１８８】
分割配置戦略を複数用意する場合には、データ配置特定情報９２２（図４参照）に、レプリカ識別子毎に選択された分割配置戦略の情報を記録する必要がある。

【０１８９】
本実施形態において、パーティショニングを行う際には、前記第１の実施の形態と比較して、データ構造変換手段１１３（図１７参照）における変換処理（図１４ステップＳ３０５）や、更新契機が「０」の場合のデータ構造の更新処理（図１３ステップＳ２０５）が異なり、指定された配置ノード先のデータ格納部を更新する点が相違している。

【０１９０】
また、データノードのアクセス処理時において、アクセス先が、パーティショニングにより、複数ノードにまたがる場合には、アクセス受付手段１１１（図１７参照）は、配置先の他のデータノードのアクセス処理手段１１２（図１７参照）に対してアクセス要求を発行する必要がある。

【０１９１】
更新処理時に、更新契機（図５（Ａ）参照）が「０」の場合、更新処理対象のレコードが格納されるデータノード全てのアクセス処理手段１１２に対してアクセス要求を発行する必要がある。

【０１９２】
参照処理についても、処理対象のレコードが格納されるデータノードの全てのアクセス処理手段１１２に要求を発行する。必要なデータノードの選択については、分散配置戦略に依存する。

【０１９３】
図１７は、本発明の第２の実施形態の構成を示す図であり、データノード１～Ｘの構成が示されている。本実施形態においては、前記第１の実施形態のアクセス受付手段１１１と相違して、アクセス受付手段１１１は、自ノード内のアクセス処理手段１１２に対して、アクセス要求を発行する際に、他ノードのアクセス処理手段１１２にも発行する場合がある。同様に、データ構造変換手段１１３は、定期的に変換処理の必要の有無を判定し、データ構造の変換を行う場合、パーティショニングされたデータを格納する他のデータノードのデータ構造変換手段１１３に対してデータ変換要求を発行する。本発明の第２の実施形態によれば、データを分割して複数のデータノードに格納することができる。

【０１９４】

＜実施形態３＞

次に、本発明の第３の実施形態について説明する。本実施形態では、データ構造管理情報９２１をアクセス負荷に応じて変更するようにしている。変更された値をシステムのデータ構造に反映することで、データ構造の設定内容（図５に示したようなレプリカ識別子毎のデータ構造の割り当て）の不適切さの修正や、システム運用後のアクセスパターンの変化などに対応可能とする。これを実現する制御パラメータの自律変更の動作について説明する。

【０１９５】

図１８は、本発明の第３の実施形態のデータノードの構成を示す図である。図１、図２、図４を参照して説明した前記第１の実施形態と比較して、本実施形態においては、履歴記録部７１と変更判定手段（変更判定部）７２が追加されている。本実施形態の各データノードのアクセス受付手段１１１（あるいは他の任意の手段において）は受け付けたアクセス要求を履歴記録部７１に記録するよう動作する。履歴記録部７１は、各テーブル識別子のレプリカ識別子毎のアクセス要求（あるいはアクセス処理内容）を記録する。

【０１９６】

履歴記録部７１は、システム全体で１つ備えた構成としてもよい。あるいは、各データノードに履歴記録部７１を備え、各データノードで個別に各テーブル識別子のレプリカ識別子毎のアクセス要求を記録していき、各データノードで個別に集められたアクセス履歴を、任意の方法で、集約する仕組みを設けてもよい。

【０１９７】

変更判定手段（変更判定部）７２は、履歴記録部７１に格納された履歴情報を用いて、データ構造を変換するか否かについて判定する。変更判定手段７２は、システム全体で１つ備えた構成としてもよいし、あるいは、各データノードで変更判定手段７２を分散して動作させ、変更判定を行うような構成としてもよい。

【０１９８】

変更判定手段７２は、構造変換が必要な際に、構造情報変更手段９１に対して、データ構造の変換処理要求を発行する。

【０１９９】

構造情報変更手段９１は、変更判定手段７２からの変換処理要求に応答して、構造情報保持部９２の情報を変更し、さらに、対象データノードのデータ管理・処理手段１１内のデータ構造変換手段１１３に対して変換処理を要求する。

【０２００】

本発明の第３の実施形態における制御パラメータの自律変更およびデータ構造の自律変換動作の流れについて、図１９、図２０、図２１を用いて説明する。

【０２０１】

＜制御動作＞

図１９は、図１８に示した本実施形態における制御動作を説明するフローチャートである。図１９の動作を、例えば定期的に行うことによって、システムのデータ構造を自律的に変更・反映することが出来る。実行周期は、任意であるが、例えば周期を長くした場合、実行中の変更処理と、整合を取る必要がある。また、周期的な実行以外にも、所定のイベント検出に応答して変更処理を行うようにしてもよい。イベントとしては、例えばシステムの任意のいずれかの構成要素により、負荷の変更を検出（例として、一部のデータノードのＣＰＵ、ディスクなどのハードウェア利用率の大きな変化など）した場合等である。

【０２０２】

図１９の動作フローは、テーブル識別子毎の構造変換処理の必要の有無の判定と、変換処理を示すものである。システムが保持管理する全てのテーブル識別子について、図１９のフローを行う必要がある。

【０２０３】

変更判定手段７２は、履歴記録部７１のアクセス履歴情報の取得を行う（ステップＳ４

０１）。
【０２０４】
次に、変更判定手段７２は、取得したアクセス履歴情報を利用して、最近の一定期間（例えば最近１日以内、あるいは最近１週間以内等）に受け付けた全てのアクセス内容が、該当テーブル識別子のいずれかのレプリカとして適したデータ構造を持っているか否かを判定する（ステップＳ４０２）
【０２０５】
ステップＳ４０２において受け付けたアクセス内容に対して、レプリカ識別子のいずれかに適したデータ構造を持っている場合には、ステップＳ４０３に進む。ここで、レプリカ識別子のいずれかに適したデータ構造を持っている場合とは、例えば、列（カラム）アクセスが必要なアクセス要求を受け付けている際に、任意のレプリカ識別子のデータ構造として、カラムストア構造を持っている場合等である。
【０２０６】
ステップＳ４０３では、変更換判定手段７２は、各レプリカ識別子が不要なデータ構造を持っているかどうか判定する。例えば、列アクセスが必要なアクセス要求が履歴として全く無いのに、カラムストア構造を多数持つ場合、不要なデータ構造といえる。
【０２０７】
不要なデータ構造が無い場合には、特に変換処理をする必要が無いため、変更判定手段７２は、フローを終了する。一方、不要なデータ構造がある場合、ステップＳ４０４に進む。
【０２０８】
ステップＳ４０４において、変更判定手段７２は、各レプリカ識別子のデータ構造と、アクセス要求量・内容から、データ構造の変更の可否の判断を行う。データ構造の変更の可否の判断は、例えば予め定義したルール等に基づいて行われる。
【０２０９】
ルールとしては、以下が挙げられる。特に制限されるものでないが、ルールは、ｉｆ＜条件＞　ｔｈｅｎ　＜アクション＞（条件成立時アクションを実行）のｉｆ－ｔｈｅｎ構造とされる。
【０２１０】
（Ｒ１）列アクセスのアクセス要求数が一定以下、且つ、行アクセスの総アクセス要求が一定数以上の場合、カラムストア構造をロウストア構造に変換する（またはその逆）。
【０２１１】
（Ｒ２）テーブル識別子に対するアクセス要求総数が一定以上の場合、レプリカ数を増やす。
【０２１２】
（Ｒ３）テーブル識別子に対し、あるカラムの値による検索クエリーが一定数以上ある場合、いずれかのレプリカ識別子にインデックスを付与する。逆にアクセスが無い場合に、インデックスを削除する。
【０２１３】
（Ｒ４）テーブル識別子に対し、リード処理要求が一定数以上ある場合に、パーティショニング数を増加する（あるいは、この逆）。
【０２１４】
（Ｒ５）テーブル識別子に対し、複数レコードにまたがる更新処理要求が一定数以上ある場合に、パーティショニング数を削減する。あるいは、パーティショニング数を「１」にする。
【０２１５】
なお、ルールは上記に制限されず、任意のものを動作させてよい。
【０２１６】
ステップＳ４０４によってデータ構造やレプリカ数を変更する必要がある場合、ステップＳ４０５へ進む。その必要が無い場合、変更判定手段７２は、フローを終了する。

【０２１７】
ステップＳ４０５において、変更判定手段７２、構造情報変更手段９１、データ構造変換手段１１３等により、データ構造を実際に変換する。レプリカを増やす場合、構造情報管理手段９のデータ構造管理情報９２１に、レプリカを増やすテーブル識別子のレコードを１つ増やし、ユニークなレプリカ識別子を付与し、その配置ノード先を決定する。配置ノードの決定は、前記第１の実施形態と同様にして行われるが、可用性レベル以上のレプリカ数を保持していれば、他の配置ノードと重複しても良い。
【０２１８】
また、レプリカは、新しいレプリカ識別子と同一のレプリカから配置ノード先へデータを複製する。
【０２１９】
ステップＳ４０５のデータ構造を変換する動作について、図２０、図２１を参照して、より詳細に説明する。簡単化のため、図２０、図２１については、レプリカ識別子は、パーティションニングされていない。以下では、図１８のデータ構造変換手段１１３の変換処理は、データ構造をＢからＣに変換する例に即して説明する。
【０２２０】
＜データ構造変換の動作＞
図２０は、本実施形態における、データ構造変換の動作を説明するフローチャートである。
【０２２１】
まず、構造情報保持部９２（図１６）のデータ構造管理情報９２１（図４）に対して、変更判定手段７２（図１６）が変更要求を発行する（ステップＳ５０１、すなわち図１９のステップＳ４０５）。これにより、構造情報変更手段９１は、変更先のデータノードＸのデータ構造変換手段１１３に対して、変換処理要求を行う。
【０２２２】
ステップＳ５０２において、変更先のレプリカ識別子のデータをもつデータノードＸでは、該当レプリカ識別子のローカル複製（局所的複製）を作成する。このローカル複製は、物理コピーではなく、ストレージによるスナップショット技術を用いてもよい。また、複製を取らず、変換元のデータとして、他ノードのレプリカ識別子のデータを用いても良い。この複製処理は、変換処理の実装方式によっては、必ずしも必要が無い。
【０２２３】
さらに、ステップＳ５０３において、構造変換処理として、データ構造変換手段１１３は、変換元のデータをデータ格納部から読み出し、変換先のデータとして異なるデータ構造として書き込む処理を行う。
【０２２４】
データ構造変換手段１１３による構造変換の完了後に、変換処理中（あるいは変換処理開始の時点で）蓄積されているデータ構造Ａのデータ格納部にデータ構造Ａのデータ構造で格納されている更新データを、変換先のデータ構造に適用する（ステップＳ５０４）。
【０２２５】
最後に、データ構造管理情報９２１（図４参照）の内容を変更し、クライアントノード６のデータアクセス手段６１１（図１参照）がアクセス要求の応答後に変換先のデータを用いるようにする（ステップＳ５０５）。
【０２２６】
データ構造管理情報９２１（図４）の変更後、変換元のデータを削除する。なお、変換元のデータは必ずしも削除しなくてもよいが、削除することで、メモリ利用効率が向上する。
【０２２７】
＜データ構造変換処理時のデータノードの処理＞
図２１は、図１８に示した本実施形態における変換処理中のデータノード内の処理を説明する図である。図１８のデータ構造変換手段１１３でデータ構造の変換処理中（ステッ

プＳ５０２－５０４）において、アクセス処理手段１１２は、アクセス要求を、データ構造Ａとデータ構造Ｂを用いて、アクセス要求を応答する。このとき、更新処理は、データ構造Ａ（Ｗｒｉｔｅ向け中間構造）に保持しておき、データ構造変換手段１１３で変換処理中は、データ構造Ｂ（Ｒｏｗ－Ｓｔｏｒｅ）への適用を行わない。

【０２２８】

データ構造変換手段１１３でのデータ構造変換処理が完了後（ステップＳ５０５）に、アクセス処理手段１１２は、Ｗｒｉｔｅ向け中間構造であるデータ構造Ａと、変換先のデータ構造Ｃ（Ｃｏｌｕｍｎ　Ｓｔｏｒｅ）を用いて、アクセス要求を処理する。

【０２２９】

なお、クライアント機能実現手段６１（図１参照）から、アクセス先のデータノードを決定する際に、データ構造変換処理中のデータノードにはアクセスせず、他のレプリカ識別子のデータを用いるようにした場合、図２１に示すように、データ構造変換処理中におけるアクセス処理手段１１２の排他処理の一部は不要になり、システム構成が簡略化される。逆に、図２１のような制御機構を具備することで、データ構造変換処理中のレプリカ識別子データでも処理を行うことが出来る。

【０２３０】

＜パーティション数の変更動作＞

図２２、図２３は、本実施形態において、パーティション数を変更する動作を説明するフローチャートである。パーティション数の変更処理は、図１９と同一のフローチャートとして表現できる。以下では、図２２について、図１９との相違点に着目して説明する。また、パーティション数だけでなく、分散戦略を変更してもよい。分散戦略の変更の一例として、例えばラウンドロビンによる分散から、任意のカラムの値範囲による分散への変更、あるいはその逆等があげられる。

【０２３１】

ステップＳ６０２（図１９のステップＳ４０２に相当）は、変更判定手段７２は、アクセス要求処理数に対し、必要性能に十分な分散数を保持しているか否かを判定する（例えば、全データをスキャンするような処理のような、データ並列の処理に対しては分散されている方が性能として有利なことが多い）。必要十分な分散数であれば、ステップＳ６０３に進む。必要十分な分散数でなければ、ステップＳ６０４に進む。

【０２３２】

ステップＳ６０３において、変更判定手段７２は、レプリカ識別子毎に不要な分割がされていないか判定する。例えば、データ並列のアクセス処理要求が少ないのに、過剰に分散配置されているレプリカ識別子が該当する。

【０２３３】

不要な分割がされていれば、ステップＳ６０４へ進み、無ければフローを終了する。

【０２３４】

ステップＳ６０４において、変更判定手段７２は、パーティション数の変更の要否判断を行う。前述したように、任意に指定されたルールに基づき、パーティション数の変更内容を決定する。変更が不要の場合には、変更判定手段７２は、フローを終了する。変更が必要な場合には、変更判定手段７２は、パーティション数を変更する（ステップＳ６０５）。ステップＳ６０５は、パーティション数を実際に変更する処理である。

【０２３５】

＜パーティション数の変更処理＞

図２３に、図２２のステップＳ６０５（変更判定手段７２によるパーティショニング数変更処理）のフローを示す。以下では、図２３について、図２０と異なる点に着目して説明する。

【０２３６】

ステップＳ７０２のローカル複製は、図２１に示したような変換処理中のアクセス要求の応答に利用するために準備する。

【０２３７】

ステップＳ７０３では、パーティション数の変更により、配置ノードが変更されるレコードについて、データを変更先のデータノードにコピーする処理である。

【０２３８】

ステップＳ７０４は、図２０のＳ５０４とほぼ同等であるが、データ構造Ａに格納されているデータ構造変換中の更新処理内容の適用先が、別のデータノードになることがある点が異なる。

【０２３９】

ステップＳ７０５は、図２０のＳ５０５とほぼ同等である。

【０２４０】

パーティショニングされたデータについて、配置先ノードを変更したり、一部のデータをディスクに書き出したり、別途用意したアーカイブストレージに格納することにより、システムの容量効率やストレージコストを低減することが出来る。

【０２４１】

例えば、図２４に示すように、注文履歴のような追記的にレコードを記録するような履歴記録型テーブル（Ａ）に対して、分割配置戦略を、時系列に決定し、古いデータ（Ｂ１、Ｂ２）を、ディスクに書き出すか（Ｃ１、Ｃ２）。あるいは別のアーカイブに書き出し、新しいデータ（Ｂ３：最も新しいパーティショニング・テーブル）のみをメモリ（Ｃ３）上に保持するようにしてもよい。

【０２４２】

本実施形態において、構造情報保持部９２のデータ配置特定情報９２２は、例えば図２５に示すようなものとなる。データ配置特定情報９２２は、テーブル識別子に関して各レプリカ識別子に対応して、配置ノード、分割配置戦略、配置物理媒体の各情報を有する。なお、図２４の履歴記録型テーブル（Ａ）は、テーブル識別子の順番で格納される。

【０２４３】

分割配置戦略として、配置戦略の情報（ラウンドロビン、カラム１の値分散、時系列等）が指定されている。

【０２４４】

データ配置特定情報９２２では、テーブル識別子“ｏｒｄｅｒｓ”のレプリカ識別子２が、時系列に配置ノード２－１０に分散配置され、配置先の物理媒体（memory、disk等）が指定されている。

【０２４５】

＜実施形態４＞

本発明の第４の実施形態としてコンシステント・ハッシングへの適用例を説明する。以下では、テーブルＡをカラムストア形式でコンシステント・ハッシング分割配置する場合の例について、図２６を用いて説明する。なお、本実施形態において、コンシステント・ハッシングで、データが配置されるデータノード（データ配置ノード）を決める処理は、図１８の変更判定手段７２で行うようにしてもよい。ノード情報は、変更判定手段７２により、構造情報保持部９２に記録される。特に制限されないが、本実施形態においては、キー値（Ｋｅｙ）と、前記キー値に対応してカラム毎に１又は複数のデータレコードを有するセットをロウ方向の単位とし、ロウの識別はキー値（Ｋｅｙ）で行われ、各カラムにカラム識別子（Ｖａｌｕｅ１、Ｖａｌｕｅ２、・・・）が付与されたテーブルに関して、前記キー値と、カラム識別子と、テーブル識別子を組み合せた文字列を引数としてハッシュ関数でハッシュ値を求め、前記ハッシュ値と、格納先ノードリスト情報から、コンシステントハッシュにより、データ配置先のデータノードを決定する。

【０２４６】

なお、ロウストア形式において、コンシステント・ハッシング分割する場合には、レコードのＫｅｙ値でハッシングして、データ配置ノードを決めればよい。Ｋｅｙ値あるいはユニーク（一義的）なレコードＩＤを用いてデータ配置ノードを決定する。

【０２４７】

図２６に模式的に示すように、コンシステント・ハッシング法においてハッシュ関数へ

引数を、
テーブル識別子＋カラム名＋Ｋｅｙ値
を組み合わせた文字列（テーブル識別子：ｔａｂｌｅＡ＋カラム識別子Ｖａｌｕｅ２＋Ｋｅｙ値：ａｃｃ）を渡し、ハッシュ値が算出される。

【０２４８】
該引数に対するハッシュ関数の出力（ハッシュ値）と、格納先ノードリスト（例えばデータノード１～４）の情報から、コンシステント・ハッシング法により、データノードを決定することが出来る。

【０２４９】
また、レコード毎にユニークなレコードＩＤを付与しておき、
テーブル識別子＋カラム名＋レコードＩＤ
をハッシュ関数に渡す引数としてもよい。

【０２５０】
図２７（Ａ）、（Ｂ）は、本実施形態におけるデータ配置ノードの記録方式について説明するための図である。カラムストア形式であるため、カラム毎にデータを記録する。外側の四角形は、データ配置ノードの記録領域の管理単位であり、例えばメモリやＨＤＤ（ハードディスクドライブ）のページに対応する。ページのサイズは任意としてよい。ページ内の任意の場所（図では末尾）に、テーブル識別子（ｔａｂｌｅＡ）とカラム名（ｖａｌｕｅ１）を指定するための管理情報を記録する。１つのカラム列全てが１つのページに収まらない場合には、他のユニットに記録する必要があるが、その他のユニットへのポインタ情報等を、この場所（記憶領域）に記録してもよい。セルの値は、ページ内の任意のアドレスに格納する。図２７（Ａ）では、ページの先頭側から順にセルの値（カラム名ｖａｌｕｅ１の各値）を記録している。

【０２５１】
また、セルの値が、どのＫｅｙに相当する情報であるかを示す情報を、別途、任意の場所に記録しておく必要がある。図２７（Ａ）では、同一ユニット内の管理情報の直前に記録しておく。そこには、Ｋｅｙの情報（あるいはユニークなレコードＩＤ）とそれがどのアドレスに格納されているかの情報（ポインタ）を記録する。情報（Ｋｅｙ：ｃｃ　＃８）は、Ｋｅｙ：ｃｃのセルの値がアドレス＃８、（Ｋｅｙ：ａｂ　＃４）は、Ｋｅｙ：ａｂのセルの値がアドレス＃４、（Ｋｅｙ：ａａ　＃０）は、Ｋｅｙ：ａａのセルの値がアドレス＃０に格納されていることを記録するものである。

【０２５２】
また、図２７（Ｂ）のように、同一テーブルの別のカラム（ｖａｌｕｅ２）の情報を別の記録管理ユニット（メモリ又はＨＤＤ）に記録するようにしてもよい。あるいは、さらに簡単な方法で分割配置としてもよい。

【０２５３】
本実施形態におけるパーティショニングの第１の例として、キー値と、前記キー値に対応してカラム毎に１又は複数のデータレコードを有するセットをロウ方向の単位とし、ロウの識別はキー値で行われ、各カラムにカラム識別子が付与されたテーブルのパーティショニング（カラムストア）を行う場合、テーブル識別子とカラム識別子とを組み合せた文字列を引数としてハッシュ関数でハッシュ値を求め、前記ハッシュ値と、格納先ノードリスト情報から、コンシステントハッシュにより、データ配置先のデータノードを決定し、カラム単位で別々のデータノードに分散配置するようにしてもよい。別々のデータノード間でパーティショニング単位に異なるデータ構造で格納してもよい。

【０２５４】
図２８は、テーブルのパーティショニングとして、テーブルのカラム毎に、データ配置ノードを分散して配置する場合を模式的に示す図である。ハッシュ関数へ与える値として、テーブル識別子とカラム名称（例えば、（ｔａｂｌｅＡ：ｖａｌｕｅ２）あるいは、（ｔａｂｌｅＡ：ｖａｌｕｅ３））を渡すだけでよい。該引数に対するハッシュ関数の出力（ハッシュ値）から格納ノードが算出される。

【０２５５】
あるいは、本実施形態におけるパーティショニングの第２の例として、キー値と、前記キー値に対応してカラム毎に１又は複数のデータレコードを有するセットをロウ方向の単位とし、ロウの識別はキー値で行われ、各カラムにカラム識別子が付与されたテーブルに関して１つのカラムを、パーティショニングする場合、テーブル識別子とカラム識別子と一義的な接尾子とを組み合せた文字列を引数としてハッシュ関数でハッシュ値を求め、前記ハッシュ値と、格納先ノードリスト情報から、コンシステントハッシュにより、データ配置先のデータノードを決定し、１つのカラムを、複数のデータノードに分散配置するようにしてもよい。配置先の複数のデータノード間でパーティショニング単位に異なるデータ構造で格納してもよい。

【０２５６】
図２９は、図２８において、テーブルの１つのカラムを二つにパーティショニングする場合を模式的に示す図である。この場合、カラムをパーティショニングするために、ハッシュ関数の引数として与える値として、テーブル識別子とカラム名称に加えて、数字等のユニークな接尾子を付与することで、複数種類のデータ配置ノード（格納ノード）を取得する。

【０２５７】
この結果、Ｋｅｙ値がａｂ、ａｃｃの場合、データ配置ノード（格納ノード）１に配置し、Ｋｅｙ値がｄｄ、ｅｅの場合、データ配置ノード（格納ノード）２に配置する。

【０２５８】
このＫｅｙ値と接尾子の組み合わせ（あるいはそれを計算できる値）を、図１８の構造情報保持部９２に格納する。また、Ｋｅｙ値が数字の場合、数値範囲毎に接尾子を指定するようにしてもよい。例えば、１－１００は識別子を０とする（結果として、格納ノード１に格納される）。このようにすることで、構造情報保持部９２への保持管理するデータ容量を削減することが出来る。

【０２５９】
なお、上記実施形態の第１、第２の例のテーブル・パーティショニングでは、カラムストア方式のパーティショニングを説明したが、ロウストア方式についても同様に適用可能である。この場合、カラム識別子の代わりにキー値等が用いられる。

【０２６０】
コンシステント・ハッシング方式において、例えば、分散ストレージシステムへ参加する複数のデータ配置ノードを、システムの動作状態に対応したグループに分け、データの書き込み要求を受けたデータ配置ノードでは、分散ストレージシステムへ参加する複数のデータ配置ノードに対して、グループごとに規定されるデータ複製数分、データの複製を作成するようにしてよい。この場合、各グループに対応して、データの複製作成数を決定し、複数のデータ配置ノードを論理的に配置したハッシュリングを辿り、グループごとの規定されるデータ複製数に達成するまで、複製先を探索し、複製先データ配置ノードのリストを作成するようにしてもよい。あるいは、複製先データ配置ノードのリストを受け、前記リストの各データ配置ノードに対して、複製命令を発行するようにしてもよい。クライアントからのデータの書き込み要求に対して複製先データ配置ノードのリストを作成し、ハッシュリング上に配置される複数のデータ配置ノードが属する所属グループに対応して、各所属グループに対応するデータ複製数のデータを複製するようにしてもよい。

【０２６１】
分散ストレージシステムやデータベースシステムを利用して企業の情報システムが実現されており、企業の業務内容の中心となるサービスを提供するシステムは「基幹系システム」あるいは「基幹系業務システム」と呼ばれ、販売や在庫管理システム、レジのＰＯＳシステム（Ｐｏｉｎｔ　ｏｆ　ｓａｌｅ　ｓｙｓｔｅｍ）等が含まれる。これら基幹系システムの情報を（時には集約して）、企業の意思決定に用いるためにデータ分析を行うシステムが、データウェアハウスとして知られている。これらのシステム（基幹系システム、データウェアハウス）では、一般的にデータに対するアクセス特性が異なるため、それ

ぞれのアクセス特性に向くように（高速処理を行うために）、データベースシステムを用意し、データ構造を特化させることが行われている。データウェアハウス・システムにおいては、例えば複数の基幹系システムからデータ（例えばトランザクション・データ等）を抽出し再構成し情報分析、意思決定のための大規模データベースを含む。基幹系システムのデータベースからデータウェアハウス・データベースへ、データの移行を行う必要があり、この工程は、ＥＴＬ（Ｅｘｔｒａｃｔ／Ｔｒａｎｓｆｏｒｍ／Ｌｏａｄ）と呼ばれている。ＥＴＬは、基幹系システムとデータウェアハウス・システム双方のデータ量の増大に伴い、高負荷になることが知られているが、本発明を適用することでデータ構造変換のボトルネックを解消し、ストレージの利用効率を高めることができる。

【０２６２】

本発明に係るデータ記憶システムは、並列データベースや並列データ処理システム、分散ストレージ、並列ファイルシステム、分散データベース、データグリッド、クラスタコンピュータに適用することができる。

【０２６３】

前記開示された実施形態の全部又は一部は、特に制限されないが、以下に記載される。

【０２６４】

（付記１）

それぞれがデータ格納部を備え、ネットワーク結合される複数のデータノードを備え、データ複製先のデータノードが、前記データノード間で、論理的には同一であるが、物理的には異なるデータ構造をそれぞれの前記データ格納部に保持する、少なくとも二つのデータノードを含む、分散ストレージシステム。

【０２６５】

(付記２)

複製先の前記データノードにおいて、目的のデータ構造への変換を複製データの受付とは非同期で行う、付記１記載の分散ストレージシステム。

【０２６６】

(付記３)

複製先の前記データノードにおいて、中間データ保持構造に前記複製データを保持して応答を返し、前記中間データ保持構造に保持されるデータ構造を、目的のデータ構造に非同期で変換する、付記２記載の分散ストレージシステム。

【０２６７】

(付記４)

予め定められたテーブル単位でデータの配置先のデータノード、配置先でのデータ構造、データ分割を可変に制御する手段を備えた付記２記載の分散ストレージシステム。

【０２６８】

(付記５)

データが配置されるデータノードを、コンシステント・ハッシングで求める手段を備えた、付記１乃至４のいずれか１に記載の分散ストレージシステム。

【０２６９】

(付記６)

データ更新時に行われるデータの複製において、前記複製先のデータノードでは、更新要求対象のデータを、それぞれ、指定されたデータベースでのデータ構造とは異なるデータ構造に変換してデータを前記データ格納部に格納し、その際、前記データノードは、更新対象のデータを、一旦、中間データ保持構造を保持して前記更新に対する応答を返し、前記更新要求とは非同期で目的のデータ構造に変換して格納する、付記１乃至５のいずれか１に記載の分散ストレージシステム。

【０２７０】

(付記７)

格納対象のデータを識別する識別子であるテーブル識別子に対応させて、複製を特定するレプリカ識別子と、前記レプリカ識別子に対応したデータ構造の種類を特定するデータ

構造情報と、指定されたデータ構造に変換して格納されるまでの時間情報である更新契機情報と、を、前記データ構造の種類の数に対応させて備えたデータ構造管理情報と、
前記テーブル識別子に対応して、前記レプリカ識別子と、前記レプリカ識別子に対応した１つ又は複数のデータ配置先のデータノード情報と、を備えたデータ配置特定情報と、
を記憶管理する構造情報保持部を有する構造情報管理装置と、
前記データ構造管理情報と前記データ配置特定情報とを参照して、更新処理及び参照処理のアクセス先を特定するデータアクセス部を備えたクライアント機能実現部と、
それぞれが前記データ格納部を備え、前記構造情報管理装置と前記クライアント機能実現部とに接続される複数の前記データノードと、
を備え、
前記データノードは、
前記クライアント機能実現部からのアクセス要求に基づき、更新処理を行う場合に、中間データ保持構造にデータを保持して前記クライアント機能実現部に応答を返すデータ管理・処理部と、
前記データ構造管理情報を参照し、指定された更新契機に応答して、前記中間データ保持構造に保持されるデータを、前記データ構造管理情報で指定されたデータ構造に変換する処理を行うデータ構造変換部と、
を備えることを特徴とする、付記１乃至６のいずれか１に記載の分散ストレージシステム。

【０２７１】

(付記８)
前記中間データ保持構造は、指定された目的のデータ構造としてデータが前記データ格納部に格納されるまでの間、前記データを保持する、付記７記載の分散ストレージシステム。

【０２７２】

(付記９)
前記クライアント機能実現部が、前記更新処理又は前記参照処理の内容に応じてアクセス先のデータノードを、前記データ構造管理情報と前記データ配置特定情報より選択する、付記７記載の分散ストレージシステム。

【０２７３】

(付記１０)
前記クライアント機能実現部は、前記構造情報管理装置の前記構造情報保持部に保持されている前記データ配置特定情報、又は、前記構造情報保持部に保持される情報をキャッシュする構造情報キャッシュ保持部に保持されているデータ配置特定情報を取得し、データ配置先のデータノードに対して、アクセス命令を発行する、付記７記載の分散ストレージシステム。

【０２７４】

(付記１１)
前記データノードは、アクセス受付部、アクセス処理部、データ構造変換部を備え、
前記データノードの前記データ格納部は、構造別データ格納部を備え、
前記アクセス受付部は、前記クライアント機能実現部からの更新要求を受け付け、前記データ配置特定情報においてレプリカ識別子に対応して指定されているデータノードに対して更新要求を転送し、
前記データノードの前記アクセス処理部は、受け取った更新要求の処理を行い、前記データ構造管理情報の情報を参照して更新処理を実行し、その際、前記データ構造管理情報の情報から、前記データノードに対する前記更新契機が零の場合、更新データを、前記データ構造管理情報に指定されるデータ構造に変換して前記構造別データ格納部を更新し、
前記更新契機が零でない場合、前記中間データ保持構造に、一旦、更新データを書き込み、処理完了を応答し、
前記アクセス受付部は、前記アクセス処理部からの完了通知と、レプリカ先のデータノ

ードの完了通知を受けると、前記クライアント機能実現部に対して応答し、
前記データ構造変換部は、前記中間データ保持構造のデータを、前記データ構造管理情報に指定されているデータ構造に変換し変換先の前記構造別データ格納部に格納する、付記７又は１０記載の分散ストレージシステム。
【０２７５】
(付記１２)
前記クライアント機能実現部は、参照系アクセスの場合、データノードに対して行われるデータアクセスに適しているデータ構造を選択し、レプリカ識別子を特定した後、アクセスすべきデータノードを算出し、選択されたデータノードに対してアクセス要求を発行し前記データノードからアクセス処理結果を受け取る、付記７記載の分散ストレージシステム。
【０２７６】
(付記１３)
前記クライアント機能実現部が、前記データノード内に配設されている、付記７記載の分散ストレージシステム。
【０２７７】
(付記１４)
前記クライアント機能実現部が、前記構造情報保持部に保持される情報をキャッシュする構造情報キャッシュ保持部を備えた付記１３記載の分散ストレージシステム。
【０２７８】
(付記１５)
前記クライアント機能実現部の前記構造情報キャッシュ保持部の構造情報と、前記構造情報管理装置の前記構造情報保持部に保持される構造情報を同期させる構造情報同期部を備えた付記１４記載の分散ストレージシステム。
【０２７９】
(付記１６)
前記データ構造管理情報が、データを複数のデータノードに分割して格納する分割数であるパーティション数をレプリカ識別子に対応して備え、
前記データ配置特定情報は、前記データ構造管理情報においてパーティション数が２以上に対応するレプリカ識別子に対応した配置ノードとして、複数のデータノードを含み、
アクセス要求を受けた前記データノードの前記アクセス受付部は、パーティショニングされたデータの配置先が複数のデータノードにまたがる場合に、前記複数のデータノードを構成する他のデータノードのアクセス処理部にアクセス要求を発行する、付記７記載の分散ストレージシステム。
【０２８０】
(付記１７)
アクセス要求を受けた前記データノードの前記データ構造変換部は、前記更新契機が零のとき、他のデータノードの前記データ構造変換部に対してアクセス要求を発行する、付記７又は１１記載の分散ストレージシステム。
【０２８１】
(付記１８)
アクセス要求の履歴を記録する履歴記録部と、
前記履歴記録部の履歴情報を用いてデータ構造の変換を行うか否かを判定する変更判定部と、
を備えた付記７記載の分散ストレージシステム。
【０２８２】
(付記１９)
前記変更判定部は、データ構造の変換が必要と判定した場合、前記構造情報管理装置の前記構造情報変更部に変換要求を出力し、
前記構造情報管理装置の前記構造情報変更部は、前記構造情報保持部の情報を変更し、

前記データノードの前記データ構造変換部に変換要求を出力し、
前記データノードの前記データ構造変換部は前記データノードの前記データ格納部に保持されるデータ構造の変換を行う、付記１８記載の分散ストレージシステム。
【０２８３】
(付記２０)
それぞれがデータ格納部を備え、ネットワーク結合される複数のデータノードを備えたシステムでの分散ストレージ方法であって、
データ複製先のデータノードの少なくとも二つのデータノードが、前記データノード間で、論理的には同一であるが、物理的には異なるデータ構造をそれぞれの前記データ格納部に保持する、分散ストレージ方法。
【０２８４】
(付記２１)
複製先の前記データノードにおいて、目的のデータ構造への変換を複製データの受付とは非同期で行う、付記２０記載の分散ストレージ方法。
【０２８５】
(付記２２)
複製先の前記データノードにおいて、中間データ保持構造に複製データを保持して応答を返し、前記中間データ保持構造に保持されるデータ構造を、目的のデータ構造に非同期で変換する、付記２１記載の分散ストレージ方法。
【０２８６】
(付記２３)
予め定められたテーブル単位でデータの配置先のデータノード、配置先でのデータ構造、データ分割を可変に制御する、付記２１記載の分散ストレージ方法。
【０２８７】
(付記２４)
データが配置されるデータノードをコンシステント・ハッシングで求める、付記２０乃至２３のいずれか１に記載の分散ストレージ方法。
【０２８８】
(付記２５)
データ更新時に行われるデータの複製において、前記複製先のデータノードでは、更新要求対象のデータを、それぞれ、指定された目的のデータベースでのデータ構造とは異なるデータ構造に変換してデータを前記データ格納部に格納し、その際、前記データノードは、更新対象のデータを一旦、中間構造を保持して前記更新に対する応答を返し、前記更新要求とは非同期で、目的のデータ構造に変換して格納する、付記２０乃至２４のいずれか１に記載の分散ストレージ方法。
【０２８９】
(付記２６)
格納対象のデータを識別する識別子であるテーブル識別子に対応させて、複製を特定するレプリカ識別子と、前記レプリカ識別子に対応したデータ構造の種類を特定するデータ構造情報と、指定されたデータ構造に変換して格納されるまでの時間情報である更新契機情報と、を前記データ構造の種類の数に対応させて備えたデータ構造管理情報と、
前記テーブル識別子に対応して、前記レプリカ識別子と、前記レプリカ識別子に対応した１つ又は複数のデータ配置先のデータノード情報と、を備えたデータ配置特定情報と、
を含む構造情報を構造情報管理部で記憶管理し、
クライアント側では、前記データ構造管理情報と前記データ配置特定情報を参照して、更新処理及び参照処理のアクセス先を特定し、
前記データノードは、
前記クライアント側からのアクセス要求に基き、更新処理を行う場合に、中間データ保持構造にデータを保持して前記クライアントに応答を返し、
前記データ構造管理情報を参照し、指定された更新契機に応じて、前記中間データ保持

構造から指定されたデータ構造に変換する、ことを特徴とする、付記２５記載の分散ストレージ方法。

【０２９０】

(付記２７)

前記データ構造管理情報が、データを複数のデータノードに分割して格納する分割数であるパーティション数を、レプリカ識別子に対応して備え、

前記データ配置特定情報は、前記データ構造管理情報においてパーティション数が２以上に対応するレプリカ識別子に対応した配置ノードとして、複数のデータノードを含み、

アクセス要求を受けた前記データノードでは、パーティショニングされたデータの配置先が複数のデータノードにまたがる場合に、前記複数のデータノードを構成する他のデータノードに対してアクセス要求を発行する、付記２６記載の分散ストレージ方法。

【０２９１】

(付記２８)

アクセス要求に履歴を記録する履歴記録部での履歴情報を用いて、データ構造の変換を行うか否かを判定し、変換が必要な場合、前記構造情報を変換し、さらに前記データノードのデータ構造を変換する、付記２６記載の分散ストレージ方法。

【０２９２】

(付記２９)

キー値と、前記キー値に対応して１又は複数のデータレコードを１又は複数のカラムに有するセットをロウ方向の単位とし、各カラムにカラム識別子が付与されたテーブルに関して、前記キー値と、前記カラム識別子と、前記テーブル識別子を組み合せた文字列を引数としてハッシュ関数でハッシュ値を求め、前記ハッシュ値と、格納先ノードリスト情報から、コンシステントハッシュにより、データ配置先のデータノードを決定する、付記５記載の分散ストレージシステム。

【０２９３】

(付記３０)

キー値と、前記キー値に対応して１又は複数のデータレコードを１又は複数のカラムに有するセットをロウ方向の単位とし、各カラムにカラム識別子が付与されたテーブルに関して、前記テーブル識別子と前記カラム識別子とを組み合せた文字列を引数としてハッシュ関数でハッシュ値を求め、前記ハッシュ値と、格納先ノードリスト情報から、コンシステントハッシュにより、データ配置先のデータノードを決定し、カラム単位で別々のデータノードに分散配置する、付記５記載の分散ストレージシステム。

【０２９４】

(付記３１)

キー値と、前記キー値に対応して１又は複数のデータレコードを１又は複数のカラムに有するセットをロウ方向の単位とし、各カラムにカラム識別子が付与されたテーブルに関して、前記テーブル識別子と前記カラム識別子と一義的な接尾子とを組み合せた文字列を引数としてハッシュ関数でハッシュ値を求め、前記ハッシュ値と、格納先ノードリスト情報から、コンシステントハッシュにより、データ配置先のデータノードを決定し、１つのカラムを、複数のデータノードに分散配置する、付記５記載の分散ストレージシステム。

【０２９５】

(付記３２)

１又は複数のデータレコードを１又は複数のカラムに有するセットをロウ方向の単位とし、各カラムにカラム識別子が付与され、レコード毎に一義的なレコード識別子が付与されたテーブルに関して、前記テーブル識別子と前記カラム識別子と前記レコード識別子を組み合せた文字列を引数としてハッシュ関数でハッシュ値を求め、前記ハッシュ値と、格納先ノードリスト情報から、コンシステントハッシュにより、データ配置先のデータノードを決定する、付記５記載の分散ストレージシステム。

【０２９６】

(付記３３)

キー値と、前記キー値に対応して１又は複数のデータレコードを１又は複数のカラムに有するセットをロウ方向の単位とし、各カラムにカラム識別子が付与されたテーブルに関して、前記キー値と、前記カラム識別子と、前記テーブル識別子を組み合せた文字列を引数としてハッシュ関数でハッシュ値を求め、前記ハッシュ値と、格納先ノードリスト情報から、コンシステントハッシュにより、データ配置先のデータノードを決定する、付記２４記載の分散ストレージ方法。

【０２９７】

(付記３４)

キー値と、前記キー値に対応して１又は複数のデータレコードを１又は複数のカラムに有するセットをロウ方向の単位とし、各カラムにカラム識別子が付与されたテーブルに関して、前記テーブル識別子と前記カラム識別子とを組み合せた文字列を引数としてハッシュ関数でハッシュ値を求め、前記ハッシュ値と、格納先ノードリスト情報から、コンシステントハッシュにより、データ配置先のデータノードを決定し、カラム単位で別々のデータノードに分散配置する、付記２４記載の分散ストレージ方法。

【０２９８】

(付記３５)

キー値と、前記キー値に対応して１又は複数のデータレコードを１又は複数のカラムに有するセットをロウ方向の単位とし、各カラムにカラム識別子が付与されたテーブルに関して、前記テーブル識別子と前記カラム識別子と一義的な接尾子とを組み合せた文字列を引数としてハッシュ関数でハッシュ値を求め、前記ハッシュ値と、格納先ノードリスト情報から、コンシステントハッシュにより、データ配置先のデータノードを決定し、１つのカラムを、複数のデータノードに分散配置する、付記２４記載の分散ストレージ方法。

【０２９９】

(付記３６)

１又は複数のデータレコードを１又は複数のカラムに有するセットをロウ方向の単位とし、各カラムにカラム識別子が付与され、レコード毎に一義的なレコード識別子が付与されたテーブルに関して、前記テーブル識別子と前記カラム識別子と前記レコード識別子を組み合せた文字列を引数としてハッシュ関数でハッシュ値を求め、前記ハッシュ値と、格納先ノードリスト情報から、コンシステントハッシュにより、データ配置先のデータノードを決定する、付記２４記載の分散ストレージ方法。

【０３００】

なお、上記の特許文献の各開示を、本書に引用をもって繰り込むものとする。本発明の全開示（請求の範囲を含む）の枠内において、さらにその基本的技術思想に基づいて、実施形態の変更・調整が可能である。また、本発明の請求の範囲の枠内において種々の開示要素（各請求項の各要素、各実施形態の各要素、各図面の各要素等を含む）の多様な組み合わせないし選択が可能である。すなわち、本発明は、請求の範囲を含む全開示、技術的思想にしたがって当業者であればなし得るであろう各種変形、修正を含むことは勿論である。

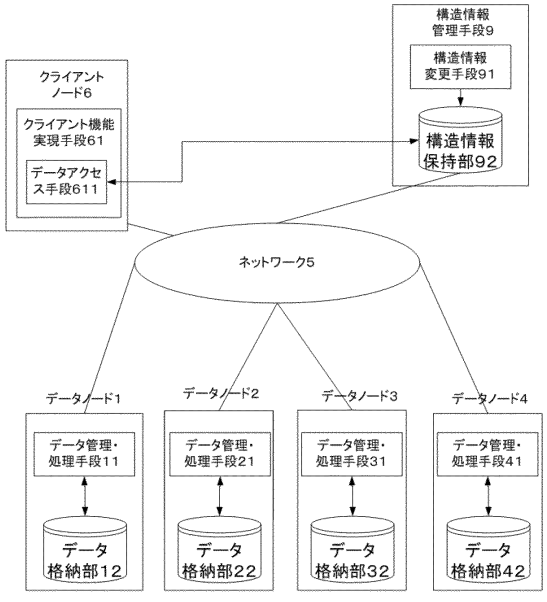
【符号の説明】

【０３０１】

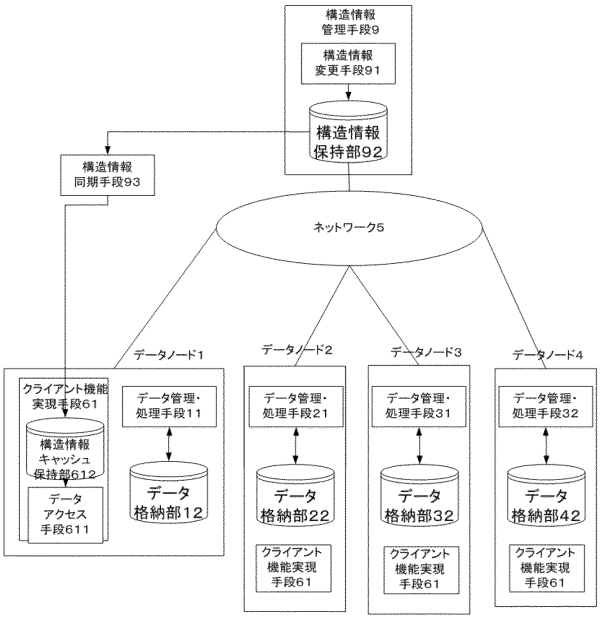
１～４　データノード
５　ネットワーク
６　クライアントノード
９　構造情報管理手段（構造情報管理装置）
１１、２１、３１、４１　データ管理・処理手段（データ管理・処理部）
１２、２２、３２、４２　データ格納部
６１　クライアント機能実現手段（クライアント機能実現部）
７１　履歴記録部
７２　変更判定手段（変更判定部）
９１　構造情報変更手段（構造情報変更部）

９２　構造情報保持部
９３　構造情報同期手段（構造情報同期部）
１０１～１０４　データノード計算機
１０１ａ　ＣＰＵ
１０１ｂ　データ記憶装置
１０１ｃ　データ転送装置
１０５　ネットワーク
１１１　アクセス受付手段（アクセス受付部）
１１２　アクセス処理手段（アクセス処理部）
１１３　データ構造変換手段（データ構造変換部）
１２１、１２２、１２３、１２Ｘ　構造別データ格納部
６１１　データアクセス手段（データアクセス部）
６１２　構造情報キャッシュ保持部
９２１　データ構造管理情報
９２２　データ配置特定情報

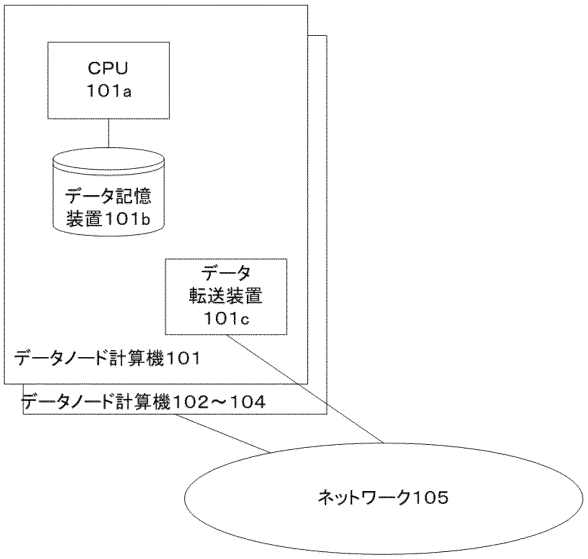
【図１】



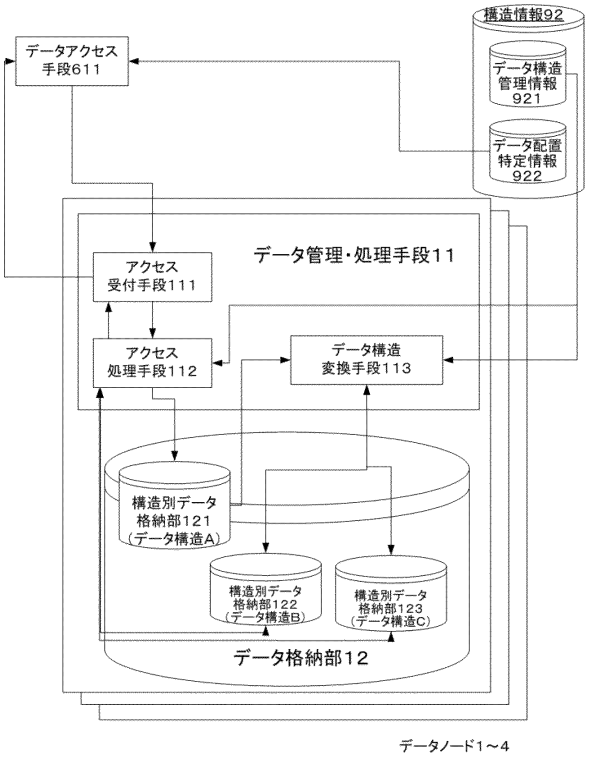
【図２】



【図３】



【図４】



【図５】

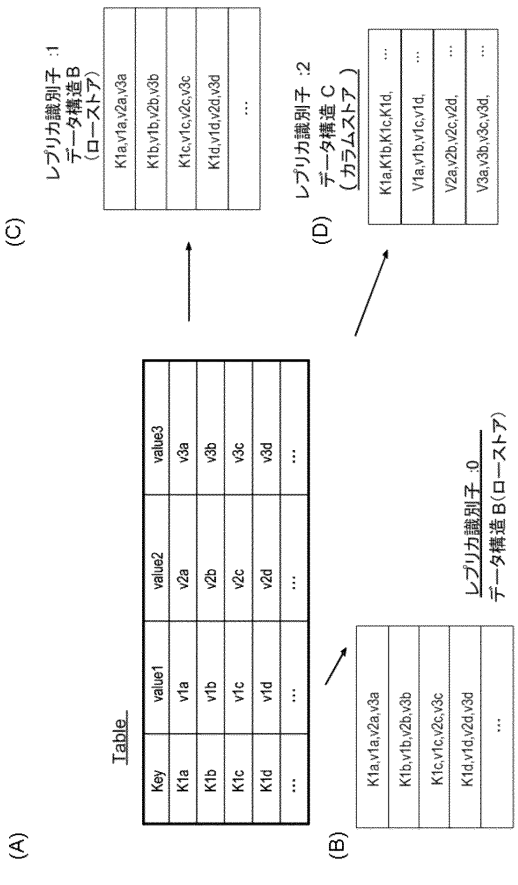
(A)

テーブル識別子	レプリカ識別子	データ構造	更新契機（sec）
Stocks	0	B	30
	1	B	60
	2	C	60
WORKERS	0	B	0
	1	B	30
	2	C	20
Shops	0	A	0
	1	B	30
	2	C	240
Salary	0	B	30
	1	B	30
	2	C	240

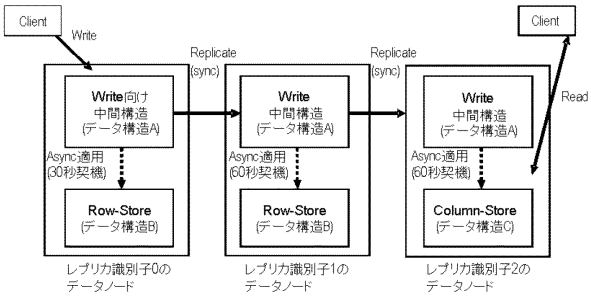
(B)

データ構造	データ格納方式の種類
A	キュー
B	ローストア
C	カラムストア
…	…

【図６】



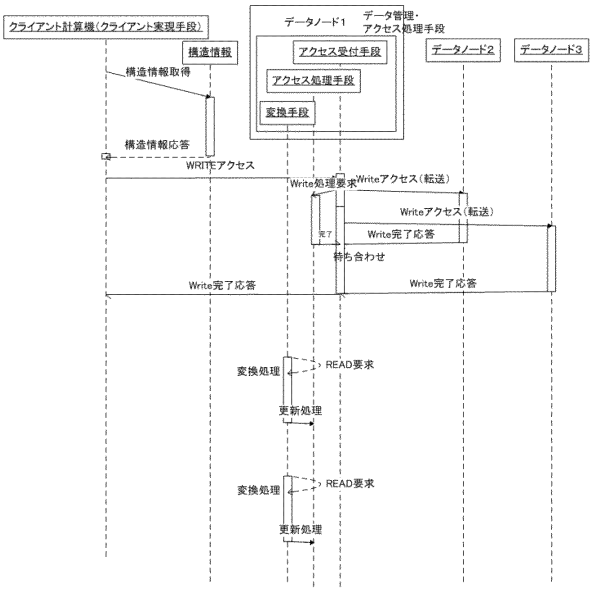
【図７】



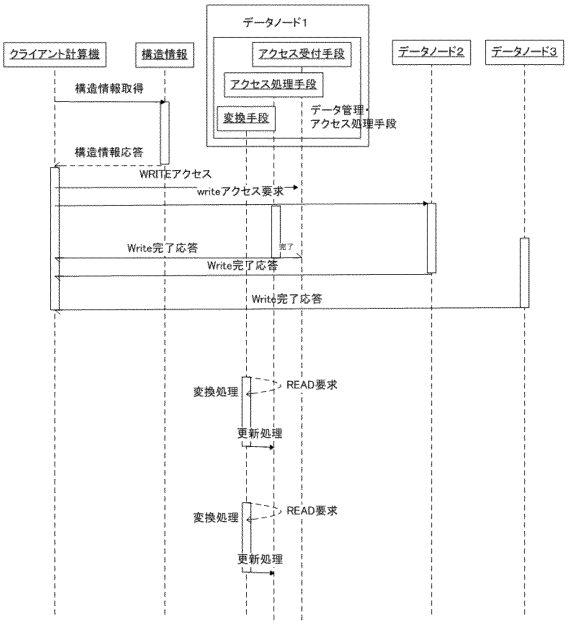
【図８】

テーブル識別子	レプリカ識別子	配置ノード
Stocks	0	1
	1	2
	2	3
WORKERS	0	4
	1	1
	2	2
Shops	0	3
	1	4
	2	1
Salary	0	2
	1	3
	2	4

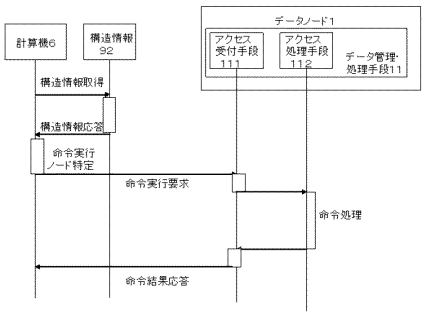
【図９】



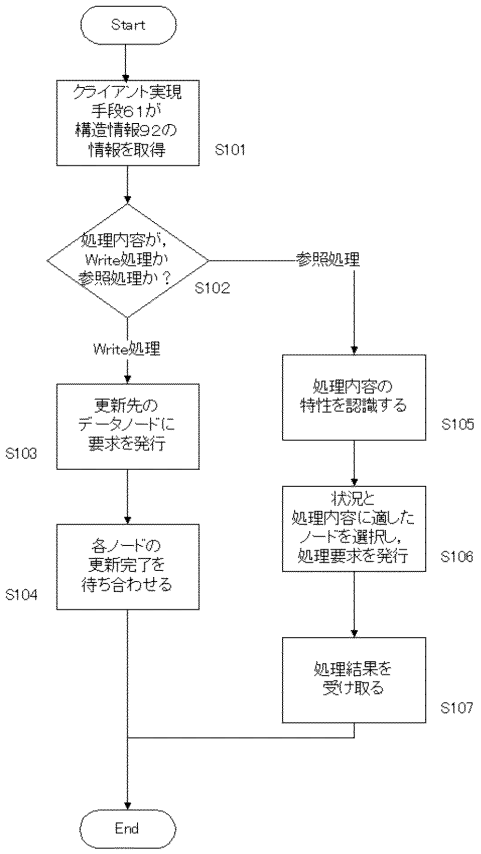
【図１０】



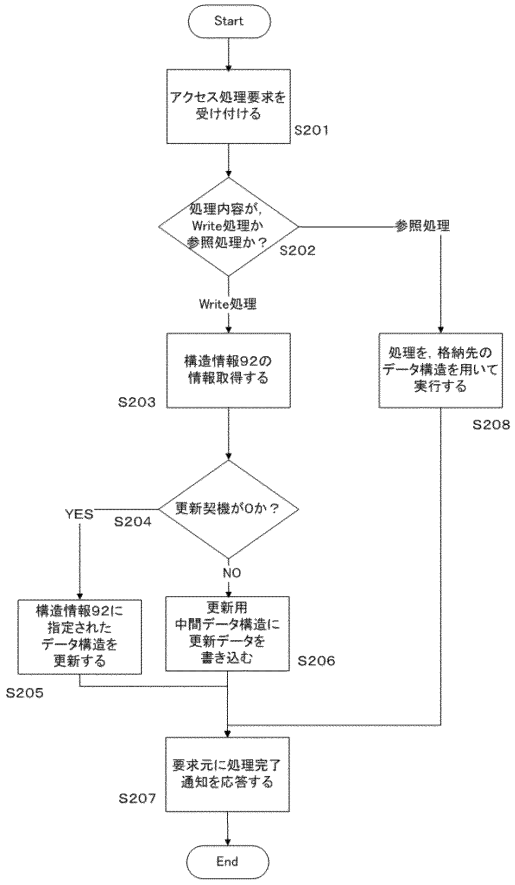
【図１１】



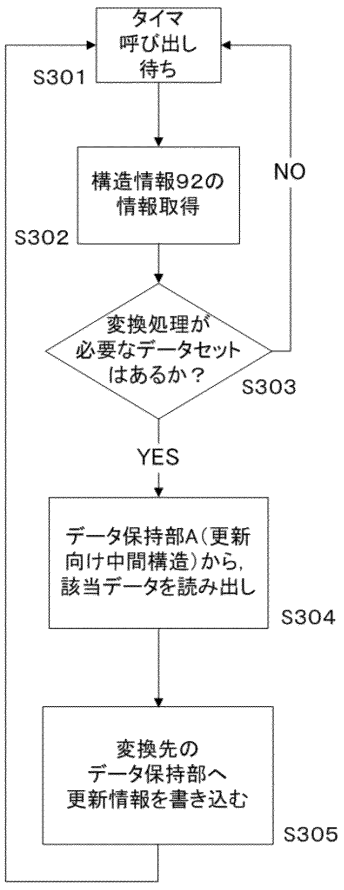
【図１２】



【図１３】



【図１４】



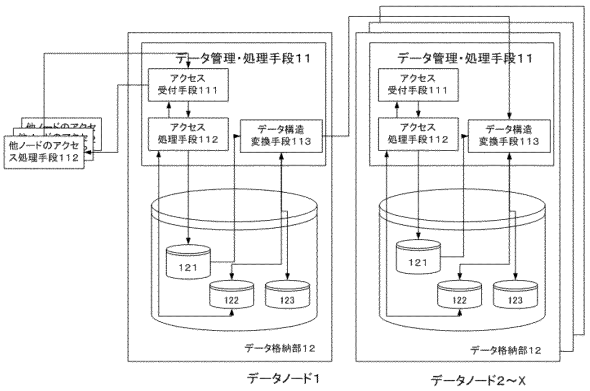
【図１５】

テーブル識別子	レプリカ識別子	パーティション数
Stocks	0	1
	1	1
	2	1
WORKERS	0	1
	1	1
	2	4
Shops	0	1
	1	1
	2	1
Salary	0	1
	1	16
	2	1

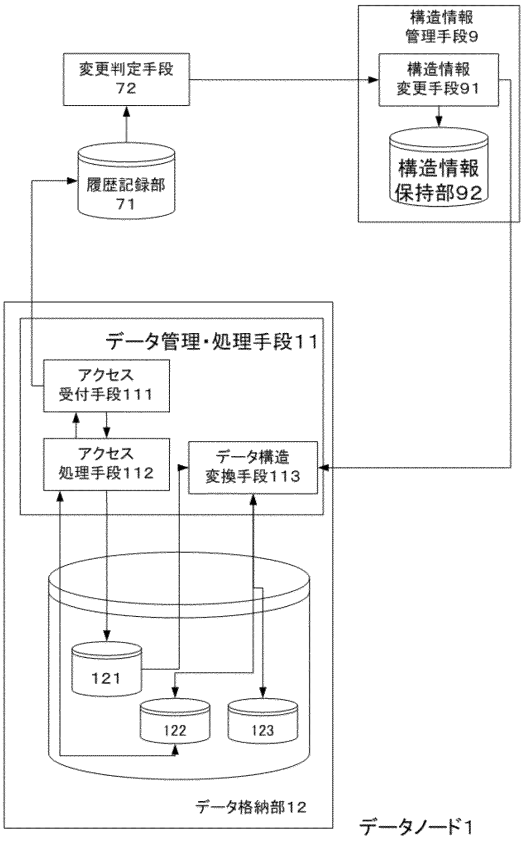
【図１６】

テーブル識別子	レプリカ識別子	配置ノード
Stocks	0	1
	1	2
	2	3
WORKERS	0	4
	1	1
	2	2,3,5,6
Shops	0	3
	1	4
	2	1
Salary	0	2
	1	3,5-15
	2	4

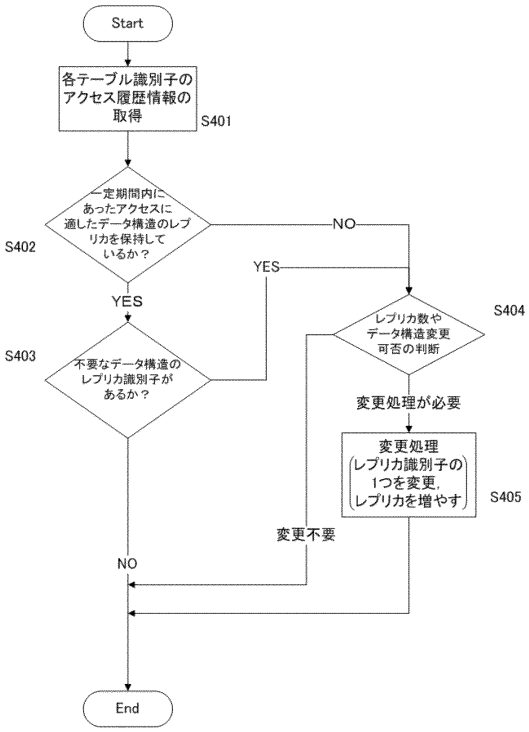
【図１７】



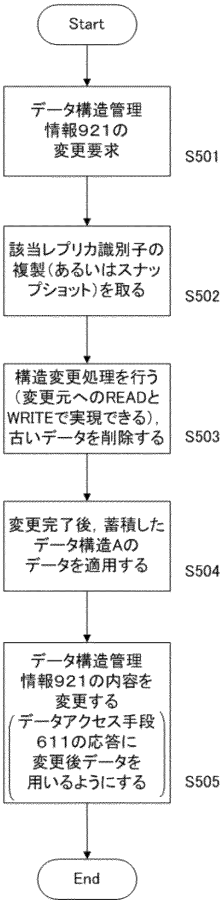
【図１８】



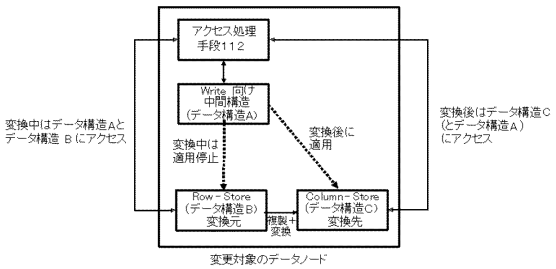
【図１９】



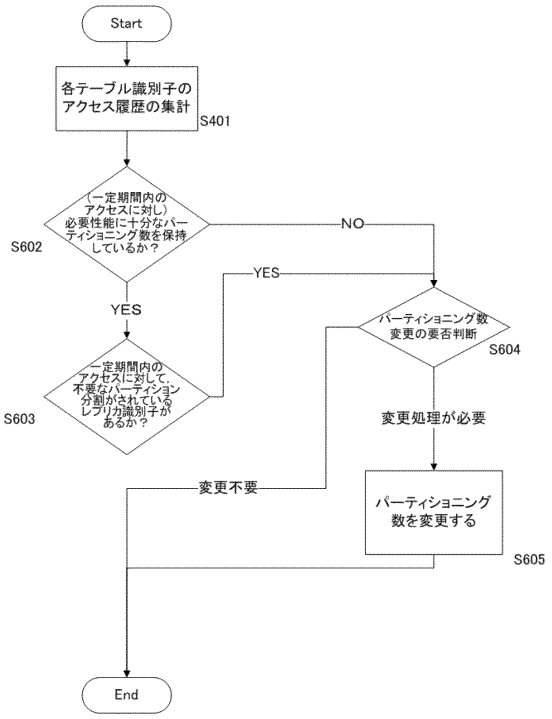
【図２０】



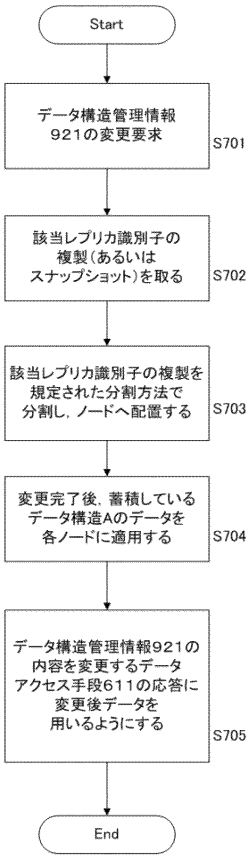
【図２１】



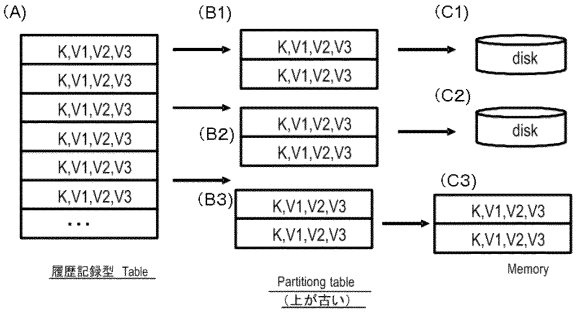
【図２２】



【図２３】



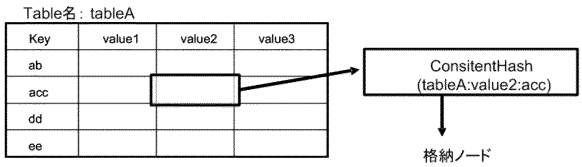
【図２４】



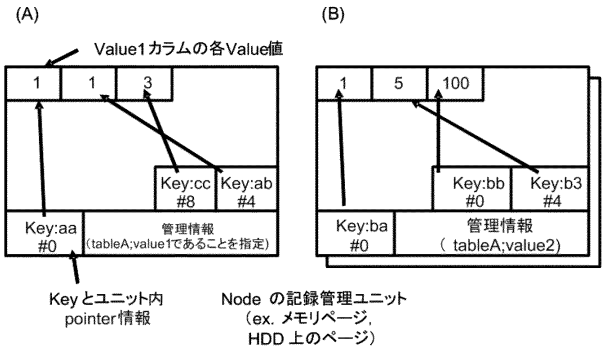
【図２５】

テーブル識別子	レプリカ識別子	配置ノード	分散配置戦略	配置物理媒体
Stocks	0	1	---	memory
	1	2	---	memory
	2	3	---	memory
WORKERS	0	4	---	memory
	1	1	---	memory
	2	2,3,5,6	roundRobin	memory
Shops	0	3	---	memory
	1	4	---	memory
	2	1	---	memory
Salary	0	2	---	memory
	1	3,5-15	カラム１の値分散	memory
	2	4	---	memory
orders	0	0	---	memory
	1	1	---	memory
	2	2-10	時系列	memory, disk,disk,…

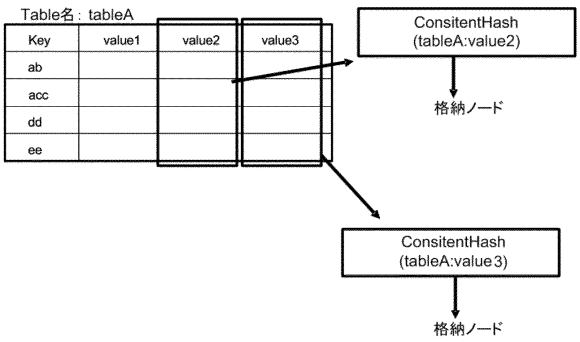
【図２６】



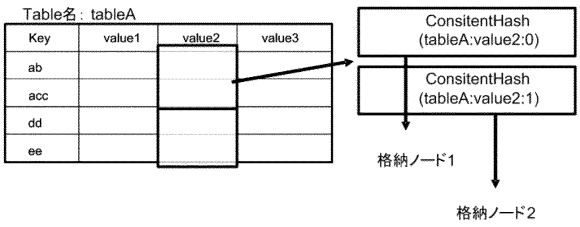
【図２７】



【図２８】



【図２９】



フロントページの続き

（出願人による申告）平成２１年度　独立行政法人新エネルギー・産業技術総合開発機構「グリーンネットワーク・システム技術研究開発プロジェクト（グリーンＩＴプロジェクト）／エネルギー利用最適化データセンタ基盤技術の研究開発／サーバの最適構成とクラウド・コンピューティング環境における進化するアーキテクチャーの開発／クラウド・コンピューティング技術」委託研究、産業技術力強化法第１９条の適用を受ける特許出願

(56)参考文献　特開２０１１－００８７１１（ＪＰ，Ａ）
特開２０１０－１４６０６７（ＪＰ，Ａ）
中村俊介 ほか１名，読み出し性能と書き込み性能を選択可能なクラウドストレージ，情報処理学会研究報告［ＣＤ－ＲＯＭ］，日本，一般社団法人情報処理学会，２０１１年　2月１５日，第2011-OS-116巻 第9号，pp.1～7
Avinash Lakshman ほか１名，Cassandra - A Decentralized Structured Storage System，ACM SIGOPS Operating Systems Review，米国，ACM，２０１０年　4月，Volume 44，Issue 2，pp.35～40

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　１２／００
Ｇ０６Ｆ　　３／０６
ＪＳＴＰｌｕｓ（ＪＤｒｅａｍＩＩＩ）