

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2016-177347
(P2016-177347A)

(43) 公開日 平成28年10月6日(2016.10.6)

(51) Int.Cl.	F I	テーマコード (参考)
G06F 17/30 (2006.01)	G06F 17/30	340D
G06F 12/00 (2006.01)	G06F 17/30	180D
	G06F 12/00	505
	G06F 12/00	513D

審査請求 未請求 請求項の数 10 O L (全 31 頁)

(21) 出願番号 特願2015-55069 (P2015-55069)
(22) 出願日 平成27年3月18日 (2015.3.18)

(71) 出願人 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 100124811
弁理士 馬場 資博
(74) 代理人 100088959
弁理士 境 廣巳
(72) 発明者 慶松 明嗣
東京都港区芝五丁目7番1号 日本電気株式会社内

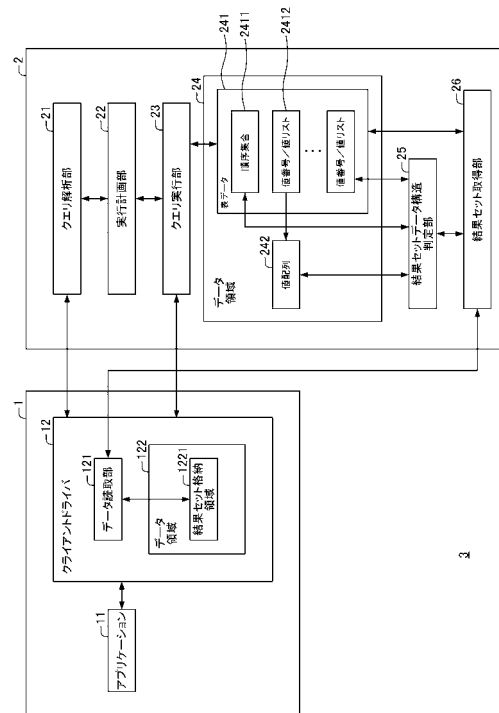
(54) 【発明の名称】 データベースシステム

(57) 【要約】

【課題】列指向のデータベースにおいて、サーバ・クライアント間のネットワークトラフィックの増大を防ぐことが難しい、という問題を解決すること。

【解決手段】データベースサーバに対して問合せを実行するクライアントと、クライアントからの問合せに応じて、各問合せに対する回答をレコードで表し、当該レコードを含む問合せの実行結果の表を送信するデータベースサーバと、を有するデータベースシステムであって、データベースサーバは、複数の列を有する実行結果の表を送信する際に、当該実行結果の表の列ごとに、当該列を表すことが出来る複数のデータ構造のうちのいずれかを用いて送信するデータ送信部を有する。

【選択図】 図2



【特許請求の範囲】**【請求項 1】**

データベースサーバに対して問合せを実行するクライアントと、前記クライアントからの問合せに応じて、各問合せに対する回答をレコードで表し、当該レコードを含む問合せの実行結果の表を送信するデータベースサーバと、を有するデータベースシステムであって、

前記データベースサーバは、複数の列を有する前記実行結果の表を送信する際に、当該実行結果の表の列ごとに、当該列を表すことが出来る複数のデータ構造のうちのいずれかを用いて送信するデータ送信部を有するデータベースシステム。

10

【請求項 2】

請求項 1 に記載のデータベースシステムであって、

前記データ送信部は、前記複数のデータ構造のそれぞれを用いた際のデータを前記クライアントに転送する際のデータ転送量に基づいて選択されたデータ構造を用いて、前記実行結果の表の列ごとにデータを送信するデータベースシステム。

【請求項 3】

請求項 2 に記載のデータベースシステムであって、

前記データ送信部は、前記複数のデータ構造のうちの前記データ転送量が最も小さくなるデータ構造を用いて、前記実行結果の表の列ごとにデータを送信するデータベースシステム。

20

【請求項 4】

請求項 1 乃至 3 のいずれかに記載のデータベースシステムであって、

前記実行結果の表の各列に対して、複数のデータ構造におけるデータ転送量を示す転送コストをそれぞれ算出し、当該算出した転送コストに基づいてデータを転送する際のデータ構造を列ごとに選択するデータ構造選択部を有し、

前記データ送信部は、前記データ構造選択部が列ごとに選択したデータ構造を用いて、列ごとにデータを送信する

を有する

データベースシステム。

30

【請求項 5】

請求項 4 に記載のデータベースシステムであって、

前記データ構造選択部は、データを値の一覧である値リストと前記値リストのうちの該当する位置を示す値番号との組み合わせで表現するデータ構造と、行番号に対応する値の配列で構成されるデータ構造と、の前記転送コストを算出し、当該算出した転送コストに基づいてデータを転送する際のデータ構造を列ごとに選択する

データベースシステム。

【請求項 6】

請求項 1 乃至 5 に記載のデータベースシステムであって、

前記クライアントは、前記データベースサーバから送信されたデータを一時的に記憶する一時データ記憶部を有し、

前記データベースサーバの前記データ送信部は、前記一時データ記憶部が記憶するデータに基づいて選択された前記複数のデータ構造のうちのいずれかを用いて送信する

データベースシステム。

40

【請求項 7】

請求項 5 に記載のデータベースシステムであって、

前記クライアントは、前記一時データ記憶部に記憶されたデータが所定の条件を満たす場合に、当該一時データ記憶部が記憶するデータをストレージ装置に移動させる

データベースシステム。

【請求項 8】

50

クライアントからの問合せに応じて、各問合せに対する回答をレコードで表し、当該レコードを含む問合せの実行結果の表を送信するデータベースサーバであって、

複数の列を有する前記実行結果の表を送信する際に、当該実行結果の表の列ごとに、当該列を表すことが出来る複数のデータ構造のうちのいずれかを用いて送信するデータ送信部を有する

データベースサーバ。

【請求項 9】

クライアントからの問合せに応じて、各問合せに対する回答をレコードで表し、当該レコードを含み複数の列項目を有する問合せの実行結果の表を送信する際に、

前記実行結果の表の列ごとに、当該列を表すことが出来る複数のデータ構造のうちのい

10

ずれかを用いて送信する
情報処理方法。

【請求項 10】

クライアントからの問合せに応じて、各問合せに対する回答をレコードで表し、当該レコードを含む問合せの実行結果の表を送信するデータベースサーバに、

複数の列を有する前記実行結果の表を送信する際に、当該実行結果の表の列ごとに、当該列を表すことが出来る複数のデータ構造のうちのいずれかを用いて送信するデータ送信手段を実現させるための

プログラム。

20

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、列指向のデータベースシステム、データベースサーバ、情報処理方法、プログラムに関する。

【背景技術】

【0002】

表の 1 列 1 列を一つのかたまりとしてデータを格納する列指向データベースが知られている。列指向データベースでは、列方向にデータをまとめている。そのため、列指向データベースは集計などの処理を高速に行うことに優れており、大量データの一括更新を行うようなバッチ処理や大量データの集計や分析などを行う場合に活用されている。

30

【0003】

また、列指向データベースの中にはFAST(Filter Array Structure)構造を持つものが知られている。FAST構造は、データを順序、位置、値などに分解して管理しており、サーバ内の処理性能に優れている。

【0004】

FAST構造を有する列指向データベースについての技術としては、例えば、特許文献 1 が知られている。特許文献 1 には、テーブル抽出部と値リスト変換部とポインタ処理部と情報ブロック管理部とを備えたCPUが記載されている。特許文献 1 によると、情報ブロック管理部が取り出した情報ブロックの値リストをテーブル抽出部が参照して、共有化すべき値リストを抽出する。続いて、値リスト変換部が、項目値の変換が必要となる値リストに項目値を挿入する。その後、ポインタ処理部がポインタ値を変換するとともに、必要な他のポインタ配列を生成する。特許文献 1 によると、このような処理により、FAST構造で表された複数の表を結合することが出来る。

40

【0005】

また、データベースにおいては、複数のデータノードにデータを分散して格納する分散ストレージシステムが知られている。分散ストレージシステムを利用とした技術としては、例えば、特許文献 2 がある。特許文献 2 には、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有する少なくとも 2 つのデータノードを含む分散ストレージシステムが記

50

載されている。特許文献2によると、このように構成することで、データの利用形態の特性が異なるアプリケーションなどに対しても迅速に応答することが可能となり、応答性能の低下などを回避することが可能となる。

【先行技術文献】

【特許文献】

【0006】

【特許文献1】特許第4428488号

【特許文献2】WO2012/121316号

【発明の概要】

【発明が解決しようとする課題】

10

【0007】

しかしながら、特許文献1に記載のようなFAST構造を利用している場合、データの整合性を確保するために膨大な値リストを送信する必要がある場合があり、その結果、ネットワークトラフィックが増加してしまうことがある、という問題が生じていた。また、特許文献2でも、同様の問題が生じる可能性があった。

【0008】

この問題に対しては、FAST構造から行指向のデータ構造へ変換してからクライアントに送信することで不必要なデータの送信を防ぐ、という対処法が知られている。しかしながら、行指向のデータ構造に変換した場合、重複するデータを送ってしまう場合がある。このように、行指向のデータ構造へ変換したとしても、ネットワークトラフィックの増大を防ぐことが難しい場合があった。

20

【0009】

以上のように、列指向のデータベースにおいては、サーバ・クライアント間のネットワークトラフィックの増大を防ぐことが難しい、という問題が生じていた。

【0010】

そこで、本発明の目的は、列指向のデータベースにおいて、サーバ・クライアント間のネットワークトラフィックの増大を防ぐことが難しい、という問題を解決するデータベースシステムを提供することにある。

【課題を解決するための手段】

【0011】

かかる目的を達成するため本発明の一形態であるデータベースシステムは、データベースサーバに対して問合せを実行するクライアントと、前記クライアントからの問合せに応じて、各問合せに対する回答をレコードで表し、当該レコードを含む問合せの実行結果の表を送信するデータベースサーバと、を有するデータベースシステムであって、

30

前記データベースサーバは、複数の列を有する前記実行結果の表を送信する際に、当該実行結果の表の列ごとに、当該列を表すことが出来る複数のデータ構造のうちのいずれかを用いて送信するデータ送信部を有する

という構成を採る。

【0012】

40

また、本発明の他の形態であるデータベースサーバは、

クライアントからの問合せに応じて、各問合せに対する回答をレコードで表し、当該レコードを含む問合せの実行結果の表を送信するデータベースサーバであって、

複数の列を有する前記実行結果の表を送信する際に、当該実行結果の表の列ごとに、当該列を表すことが出来る複数のデータ構造のうちのいずれかを用いて送信するデータ送信部を有する

という構成を採る。

【0013】

また、本発明の他の形態である情報処理方法は、

クライアントからの問合せに応じて、各問合せに対する回答をレコードで表し、当該レ

50

コードを含み複数の列項目を有する問合せの実行結果の表を送信する際に、
前記実行結果の表の列ごとに、当該列を表すことが出来る複数のデータ構造のうちのいずれかを用いて送信する
という構成を採る。

【0014】

また、本発明の他の形態であるプログラムは、
クライアントからの問合せに応じて、各問合せに対する回答をレコードで表し、当該レコードを含む問合せの実行結果の表を送信するデータベースサーバに、
複数の列を有する前記実行結果の表を送信する際に、当該実行結果の表の列ごとに、当該列を表すことが出来る複数のデータ構造のうちのいずれかを用いて送信するデータ送信手段を実現させるためのプログラムである。

10

【発明の効果】

【0015】

本発明は、以上のように構成されることにより、列指向のデータベースにおいて、サーバ・クライアント間のネットワークトラフィックの増大を防ぐことが難しい、という問題を解決するデータベースシステムを提供することが可能となる。

【図面の簡単な説明】

【0016】

【図1】本発明の第1の実施形態に係るデータベースシステムの全体の構成を示すブロック図である。

20

【図2】本発明の第1の実施形態に係るクライアントとデータベースサーバとの構成を示すブロック図である。

【図3】FAST構造を説明するための図である。

【図4】結果セットデータ構造判定部が値番号/値リストのデータ構造を選択した場合に行う値番号の再計算処理を説明するための図である。

【図5】結果セットデータ構造判定部が値配列のデータ構造を選択した場合に行う値配列の作成処理を説明するための図である。

【図6】データベースシステムの全体的な動作を説明するためのフローチャートである。

【図7】データベースサーバに格納されている表の一例を示す図である。

【図8】クエリの実行結果の表の一例を示す図である。

30

【図9】結果セットデータ構造判定部が行う転送コストの算出及びデータ構造の選択を説明するための図である。

【図10】結果セットデータ構造判定部が行う転送コストの算出及びデータ構造の選択を説明するための図である。

【図11】結果セット取得部が付加するヘッダ情報を説明するための図である。

【図12】結果セット取得部が付加するヘッダ情報を説明するための図である。

【図13】結果セット取得部が付加するヘッダ情報を説明するための図である。

【図14】データ読取部により行われるデータへのアクセスの仕方を説明するための図である。

【図15】データ読取部により行われるデータへのアクセスの仕方を説明するための図である。

40

【図16】データベースシステムがクエリの実行を行う際の動作の一例を示すシーケンス図である。

【図17】クライアントが結果セットの取得を行う際の動作の一例を示すシーケンス図である。

【図18】データベースシステムがクエリを終了する際の動作の一例を示すシーケンス図である。

【図19】本発明の第2の実施形態に係るデータベースシステムの概要を説明するための図である。

【図20】本発明の第2の実施形態に係るクライアントとデータベースサーバとの構成を

50

示すブロック図である。

【図 2 1】結果セット取得部が付加するヘッダ情報を説明するための図である。

【図 2 2】本発明の第 2 の実施形態に係るデータベースシステムの動作の一例を示すシーケンス図である。

【図 2 3】本発明の第 2 の実施形態に係る結果セットデータ構造判定部が値リスト I D の確認結果に応じた処理を行う際の動作の一例を示すフローチャートである。

【図 2 4】本発明の第 3 の実施形態に係るクライアントとデータベースサーバとの構成を示すブロック図である。

【図 2 5】値リスト永続化表の構成の一例を示す図である。

【図 2 6】本発明の第 4 の実施形態に係るデータベースシステムの構成の概要を示す概略ブロック図である。

【発明を実施するための形態】

【0017】

[第 1 の実施形態]

図 1 は、データベースシステム 3 の全体の構成を示すブロック図である。図 2 は、クライアント 1 とデータベースサーバ 2 との構成を示すブロック図である。図 3 は、FAST 構造（順序集合 2 4 1 1 と値番号 / 値リスト 2 4 1 2）を説明するための図である。図 4 は、結果セットデータ構造判定部 2 5 が値番号 / 値リスト 2 4 1 2 のデータ構造を選択した場合に行う値番号の再計算処理を説明するための図である。図 5 は、結果セットデータ構造判定部 2 5 が値配列 2 4 2 のデータ構造を選択した場合に行う値配列 2 4 2 の作成処理を説明するための図である。図 6 は、データベースシステム 3 の全体的な動作を説明するためのフローチャートである。図 7 は、データベースサーバ 2 のデータ領域 2 4 に格納されている表の一例を示す図である。図 8 は、クエリの実行結果の表の一例を示す図である。図 9、10 は、結果セットデータ構造判定部 2 5 が行う転送コストの算出及びデータ構造の選択を説明するための図である。図 11 ~ 13 は、結果セット取得部 2 6 が付加するヘッダ情報を説明するための図である。図 14、15 は、データ読取部 1 2 1 により行われるデータへのアクセスの仕方を説明するための図である。図 16 は、データベースシステム 3 がクエリの実行を行う際の動作の一例を示すシーケンス図である。図 17 は、クライアント 1 が結果セットの取得を行う際の動作の一例を示すシーケンス図である。図 18 は、データベースシステムがクエリを終了する際の動作の一例を示すシーケンス図である。

【0018】

図 1 で示すように、本発明の第 1 の実施形態では、クライアント 1 とデータベースサーバ 2 とを有するデータベースシステム 3 について説明する。クライアント 1 は、データベースサーバ 2 に対してクエリ（問合せ、処理要求）を送信する。データベースサーバ 2 は、クライアント 1 からクエリを受信すると、クエリを実行して、各問合せに対する回答をレコード（各行）で表すクエリの実行結果の表を FAST (Filter Array Structure) 構造で生成する。そして、データベースサーバ 2 は、クエリの実行結果の表をクライアント 1 に対して送信する。後述するように、本実施形態におけるデータベースサーバ 2 は、複数の列を有するクエリの実行結果の表を送信する際に、実行結果の表の各列に対して複数のデータ構造における転送コストをそれぞれ算出する。そして、データベースサーバ 2 は、算出結果に基づいて列ごとにデータ構造を選択し、当該選択したデータ構造で列ごとにクエリの実行結果の表を送信する。

【0019】

なお、クライアント 1 とデータベースサーバ 2 とは、例えばネットワークを介して接続されており、互いに通信可能なよう構成されている。また、データベースサーバ 2 と接続されたクライアント 1 の数は、1 つでも構わないし、2 つ以上の複数でも構わない。

【0020】

クライアント 1 は、情報処理装置である。クライアント 1 は、図示しない演算装置 (CPU: Central Processing Unit) と、記憶装置と、を有してい

10

20

30

40

50

る。クライアント 1 は、記憶装置が有するプログラムを CPU が実行することで、後述するアプリケーション 1 1 とクライアントドライバ 1 2 とを実現するよう構成されている。

【0021】

図 2 を参照すると、本実施形態におけるクライアント 1 は、アプリケーション 1 1 とクライアントドライバ 1 2 とを有している。

【0022】

アプリケーション 1 1 は、データベースサーバ 2 を利用する任意のアプリケーションである。アプリケーション 1 1 は、例えば、Java (登録商標) アプリケーションである。なお、アプリケーション 1 1 は、Java アプリケーション以外であっても構わない。

【0023】

クライアントドライバ 1 2 は、アプリケーション 1 1 とデータベースサーバ 2 との間のインターフェースである。クライアントドライバ 1 2 は、例えば、アプリケーション 1 1 からの要求に応じてデータベースサーバ 2 に対してクエリを送信したり、結果セット取得要求を送信したりする。また、クライアントドライバ 1 2 は、例えば、データベースサーバ 2 からクエリの結果の表を受信して、アプリケーション 1 1 に渡す。このように、クライアントドライバ 1 2 は、アプリケーション 1 1 がデータベースサーバ 2 を利用する際に利用される。クライアントドライバ 1 2 は、例えば、JDBC/ODBC (Open Database Connectivity) ブリッジドライバである。なお、クライアントドライバ 1 2 は、JDBC/ODBC 以外のドライバであっても構わない。

【0024】

図 2 で示すように、クライアントドライバ 1 2 は、データ読取部 1 2 1 とデータ領域 1 2 2 とを有している。

【0025】

データ読取部 1 2 1 は、データベースサーバ 2 から、クエリの実行結果の表を列ごとに受信する。例えば、データ受取部 1 2 1 は、データベースサーバ 2 から TCP/IP (Transmission Control Protocol/Internet Protocol) で転送されたデータを受信する。すると、データ受取部 1 2 1 は、当該受信したデータのヘッダ情報を参照して、受信したデータのデータ構造を識別する。そして、データ受取部 1 2 1 は、受信したデータをデータ領域 1 2 2 (の後述する結果セット格納領域 1 2 2 1) に格納する。

【0026】

データ領域 1 2 2 は、半導体メモリなどの記憶装置である。データ領域 1 2 2 には、結果セット格納領域 1 2 2 1 が設けられている。上記のように、結果セット格納領域 1 2 2 1 には、データ読取部 1 2 1 がデータベースサーバ 2 から受信したデータが記憶される。結果セット格納領域 1 2 2 1 は、例えば、クライアントドライバ 1 2 により問合せ (クエリ) ごとに生成され、クライアントドライバ 1 2 により問合せの終了と同時に破棄される。

【0027】

データベースサーバ 2 は、情報処理装置である。データベースサーバ 2 は、図示しない演算装置 (CPU) と、記憶装置と、を有している。データベースサーバ 2 は、記憶装置が有するプログラムを CPU が実行することで、後述する各機能 (各部) を実現するよう構成されている。

【0028】

図 2 を参照すると、データベースサーバ 2 は、クエリ解析部 2 1 と、実行計画部 2 2 と、クエリ実行部 2 3 と、データ領域 2 4 と、結果セットデータ構造判定部 2 5 と、結果セット取得部 2 6 と、を有している。

【0029】

クエリ解析部 2 1 は、クライアントドライバ 1 2 から送信されたクエリを構文解析する。つまり、クエリ解析部 2 1 は、クライアントドライバ 1 2 からクエリを受信する。そして、クエリ解析部 2 1 は、受信したクエリの構文解析を実行する。その後、クエリ解析部

10

20

30

40

50

2 1 は、解析結果を実行計画部 2 2 へ送信する。

【 0 0 3 0 】

実行計画部 2 2 は、クエリ解析部 2 1 が構文解析したクエリをどのような順序、方法で行えば効率的であるかを判定し、その実行計画を作成する。そして、実行計画部 2 2 は、作成した実行計画をクエリ実行部 2 3 に送信する。

【 0 0 3 1 】

クエリ実行部 2 3 は、実行計画部 2 2 で作成した実行計画に従って、データ領域 2 4 に対してデータ操作を行う。クエリ実行部 2 3 は、クエリの実行結果の表をデータ領域 2 4 に記憶することになる。

【 0 0 3 2 】

データ領域 2 4 は、半導体メモリやディスク装置などの記憶装置である。データ領域 2 4 には、データベースサーバ 2 のデータベースに登録されている表や、データベースサーバ 2 に対する問合せの結果（クエリ実行部 2 3 によるクエリの実行結果の表）が、表データ 2 4 1 として格納されている。

【 0 0 3 3 】

表データ 2 4 1 は、順序集合 2 4 1 1 と複数の値番号 / 値リスト 2 4 1 2 とから構成されている。このように、本実施形態におけるデータベースサーバ 2 は、表データを F A S T 構造で保管している。

【 0 0 3 4 】

順序集合 2 4 1 1 は、値番号 / 値リスト 2 4 1 2 のうちの参照する値番号の位置を示している。例えば、図 3 で示す順序集合 2 4 1 1 のうちの添え字 (i d x) 「 0 」の順序集合「 1 」は、値番号 / 値リスト 2 4 1 2 のうちの添え字「 1 」を参照することを示している。つまり、行番号 (順序集合の添え字) に対応する順序集合 2 4 1 1 の要素は、値番号の添え字に該当することになる。

【 0 0 3 5 】

値番号 / 値リスト 2 4 1 2 は、データを値の一覧である値リストと前記値リストのうちの該当する位置を示す値番号との組み合わせで表現している。例えば、図 3 で示す値番号 / 値リスト 2 4 1 2 のうちの添え字「 1 」の値番号「 2 」は、値番号 / 値リスト 2 4 1 2 のうちの値リストの添え字「 2 」を参照する (値リスト = Z) ことを示している。このように、値番号の要素は値リストの添え字を示している。

【 0 0 3 6 】

なお、図 2 では 1 つの順序集合 2 4 1 1 を示したが、順序集合 2 4 1 1 は複数あっても構わない。

【 0 0 3 7 】

また、データ領域 2 4 には、値配列 2 4 2 を格納することが出来る。値配列 2 4 2 は、行番号に対応する値の配列で構成されるデータである。値配列 2 4 2 は、結果セットデータ構造判定部 2 5 により順序集合 2 4 1 1 と値番号 / 値リスト 2 4 1 2 から生成される。値配列 2 4 2 の詳細については後述する。

【 0 0 3 8 】

結果セットデータ構造判定部 2 5 は、結果セット (クエリ実行部 2 3 により実行されたクエリの実行結果の表) の各列に対して、複数のデータ構造のそれぞれを用いた場合のデータ転送量を示す転送コストをそれぞれ算出する。そして、結果セットデータ構造判定部 2 5 は、算出した転送コストに基づいて、結果セットの各列に対してどのデータ構造で転送するかをそれぞれ判定する。

【 0 0 3 9 】

具体的には、結果セットデータ構造判定部 2 5 は、結果セットの各列に対して、値番号 / 値リスト 2 4 1 2 及び値配列 2 4 2 の総バイト数を見積もり、その値を転送コストとする。つまり、結果セットデータ構造判定部 2 5 は、データ構造として値番号 / 値リスト 2 4 1 2 を用いた場合の転送コストと、データ構造として値配列 2 4 2 を用いた場合の転送コストと、を算出する。そして、結果セットデータ構造判定部 2 5 は、転送コストの少な

10

20

30

40

50

い方のデータ構造を選択する。

【0040】

また、結果セットデータ構造判定部25は、転送する際に用いるデータ構造を選択すると、当該選択したデータ構造に応じた処理を行う。

【0041】

例えば、結果セットデータ構造判定部25は、値番号/値リスト2412を選択した場合、値番号の添え字(index)を行番号に対応させる処理を行う。つまり、結果セットデータ構造判定部25は、図4で示すように、順序集合2411の要素を添え字としたときの値番号の要素を、新しい値番号(新値番号)の要素となるように、再計算する。例えば、図4を参照すると、添え字「0」の順序集合2411の要素「5」を添え字としたときの値番号の要素は「1」である。そこで、結果セットデータ構造判定部25は、添え字「0」の新値番号の要素を「1」に再計算する。このような処理を繰り返して、結果セットデータ構造判定部25は、値番号の添え字を行番号に対応させる。

10

【0042】

このように、結果セットデータ構造判定部25は、値番号/値リスト2412を選択した場合、値番号の添え字(index)を行番号に対応させる処理を行う。このような処理を行うことで、転送する結果セットに順序集合2411を含める必要がなくなることになる。

【0043】

また、結果セットデータ構造判定部25は、値配列242を選択した場合、対応する値番号/値リスト2412に基づいて、値配列242を作成する。例えば、図5を参照すると、結果セットデータ構造判定部25は、行番号に対応する順序集合2411の要素を値番号の添え字としたときの値番号の要素を求める。次に、求めた値番号を値リストの添え字としたときの値リストの要素を求める。ここで、値リストの各要素は実際のデータを意味している。そのため、結果セットデータ構造判定部25は、上記求めた値リストの要素を値配列242に追加していくことで、値配列242を作成する。

20

【0044】

このように、結果セットデータ構造判定部25は、値配列242を選択した場合、行番号が添え字に対応するように値リストの各要素を追加することで、値配列242を作成する。

30

【0045】

結果セット取得部26は、クライアントドライバ12から結果セット取得要求を受け付ける。また、結果セット取得部26は、クライアントドライバ12から受け付けた結果セット取得要求に応じて、クライアントドライバ12に対して結果セットを転送する。

【0046】

具体的には、結果セット取得部26は、結果セット取得要求に応じて、結果セットを列ごとに送信する。また、結果セット取得部26は、結果セットを列ごとに送信する際に、結果セットデータ構造判定部25が選択したデータ構造の識別情報を含むヘッダ情報を付加する。

【0047】

以上が、本実施形態におけるデータベースシステム3の構成についての説明である。次に、上記構成のデータベースシステム3が行う処理について具体例を挙げて説明する。

40

【0048】

図6で示すように、データベースシステム3の処理は、大きく、クエリの実行(ステップS001)、結果セットの取得(ステップS002)、クエリの終了(ステップS003)の3つの部分に分けることが出来る。以下、各部分の処理について説明する。

【0049】

まず、クエリの実行について説明する。この処理は、一般的なリレーショナルデータベースシステムと同様の処理となる。

【0050】

50

つまり、クライアントドライバ 12 が送信したクエリをクエリ解析部 21 が受信すると、クエリ解析部 21 は、受信したクエリの構文解析を実行する。その後、クエリ解析部 21 は、解析結果を実行計画部 22 へ送信する。続いて、実行計画部 22 は、クエリ解析部 21 が構文解析したクエリをどのような順序、方法で行えば効率的であるかを判定し、その実行計画を作成する。そして、実行計画部 22 は、作成した実行計画をクエリ実行部 23 に送信する。その後、クエリ実行部 23 は、実行計画部 22 で作成した実行計画に従ってクエリを実行する。これにより、クエリ実行部 23 は、クエリに対する実行結果の表を作成してデータ領域 24 の表データ 241 に格納する。なお、クエリ実行部 23 が作成する実行結果の表は、FAST 構造である。

【0051】

具体的に、例えば、データベースサーバ 2 の表データ 241 に図 7 で示す「顧客表」が格納されているとする。つまり、図 7 で示すように、顧客 ID、姓、名、都道府県、登録日、更新日、などの列を有する表が 100 万件登録されていたとする。そして、この時に、下記式で示すクエリを実行したとする。

```
SELECT 顧客ID, 都道府県 FROM 顧客表 WHERE 登録日<'2009-01-01';
```

【0052】

その結果、クエリ実行部 23 は、例えば、図 8 で示すような実行結果の表をデータ領域 24 の表データ 241 に FAST 構造で格納する。つまり、例えば、「顧客表」に登録日が 2009 年 1 月 1 日より前のデータが 5000 件ある場合、クエリ実行部 23 は、0 から 4999 までを示す順序集合を格納する。この際、顧客 ID 列と、都道府県列とは、元の表のままであるため、元の顧客表のデータを参照することになる。

【0053】

次に、結果セットの取得について説明する。

【0054】

結果セット取得部 26 がクライアントドライバ 12 から結果セット取得要求を受信すると、結果セット取得部 26 は、クライアントドライバ 12 から受け付けたクエリに対する実行結果の表を取得する。

【0055】

結果セットデータ構造判定部 25 は、結果セット取得部 26 が取得した実行結果の表の各列に対して、値番号/値リスト 2412 の転送コストと、値配列 242 の転送コストとの算出を行う。そして、結果セットデータ構造判定部 25 は、転送コストが小さいデータ構造を選択して、当該選択したデータ構造に応じた処理を行う。

【0056】

例えば、結果セットデータ構造判定部 25 は、下記式を用いて値番号/値リスト 2412 の転送コストと、値配列 242 の転送コストと、を算出する。

[値番号/値リストのデータ構造の転送コスト]

値番号の型のバイト数 × 順序集合の要素数 + 値リストの総バイト数

[値配列のデータ構造の転送コスト]

順序集合の要素数 × 値リストの総バイト数 / 値リストの要素数

【0057】

そして、結果セットデータ構造判定部 25 は、転送コストが小さいデータ構造を選択して、選択したデータ構造に応じた処理を行う。

【0058】

例えば、値番号/値リスト 2412 を選択した場合、結果セットデータ構造判定部 25 は、値番号の添え字 (i d x) を行番号に対応させる処理を行う。つまり、結果セットデータ構造判定部 25 は、図 4 で示すように、順序集合 2411 の要素を添え字としたときの値番号の要素を、新しい値番号 (新値番号) の要素となるように、再計算する。また、値配列 242 を選択した場合、結果セットデータ構造判定部 25 は、対応する値番号/値リスト 2412 に基づいて、値配列 242 を作成する。つまり、結果セットデータ構造判定部 25 は、図 5 で示すように、行番号が添え字に対応するように値リストの各要素を追

10

20

30

40

50

加することで、値配列 2 4 2 を作成する。

【 0 0 5 9 】

このようにして、結果セットデータ構造判定部 2 5 は、転送コストが小さいデータ構造を選択するとともに所定の処理を行う。

【 0 0 6 0 】

そして、結果セット取得部 2 6 は、値番号 / 値リスト 2 4 1 2 として選択された列の値リストをクライアントドライバ 1 2 に送信する。この際、結果セット取得部 2 6 は、転送するデータが値リストであるという値リスト識別子及び参照元の表の値リストが一意に識別できる値リスト ID をヘッダ情報として付加してクライアントドライバ 1 2 に値リストを転送する (図 1 1 参照) 。

10

【 0 0 6 1 】

その後、クライアントドライバ 1 2 のデータ読取部 1 2 1 は、データベースサーバ 2 から転送されたデータのヘッダ情報を読み取り、値リストと識別したものをデータ領域 1 2 2 の結果セット格納領域 1 2 2 1 に格納する。

【 0 0 6 2 】

上記処理を、実行結果の表の一例として図 8 で示す実行結果の表がある場合について、より具体的に説明する。この場合、結果セットデータ構造判定部 2 5 は、顧客 ID の列に対して、例えば下記のような転送コストの算出を行う (図 9 参照) 。

[値番号 / 値リストのデータ構造の転送コスト]

値番号の型のバイト数 × 順序集合の要素数 + 値リストの総バイト数
= 5 0 0 0 × 4 + 8 0 0 万 = 8 0 2 万

20

[値配列のデータ構造の転送コスト]

順序集合の要素数 × 値リストの総バイト数 / 値リストの要素数
= 5 0 0 0 × 8 0 0 万 / 1 0 0 万 = 4 万

【 0 0 6 3 】

上記転送コストの算出結果からすると、値配列 2 4 2 のデータ構造の転送コストの方が値番号 / 値リスト 2 4 1 2 のデータ構造の転送コストよりも転送コストが小さい。そのため、結果セットデータ構造判定部 2 5 は、図 9 で示すように、値配列 2 4 2 のデータ構造を選択して、値番号 / 値リスト 2 4 1 2 に基づいて値配列 2 4 2 にデータ構造の変換を行う。そして、結果セットデータ構造判定部 2 5 は、変換した値配列 2 4 2 をデータ領域 2 4 2 に格納する。

30

【 0 0 6 4 】

一方、図 8 で示す実行結果の表の場合、結果セットデータ構造判定部 2 5 は、都道府県の列に対して、例えば下記のような転送コストの算出を行う (図 1 0 参照) 。

[値番号 / 値リストのデータ構造の転送コスト]

値番号の型のバイト数 × 順序集合の要素数 + 値リストの総バイト数
= 5 0 0 0 × 1 + 3 3 7 = 約 5 0 0 0

[値配列のデータ構造の転送コスト]

順序集合の要素数 × 値リストの総バイト数 / 値リストの要素数
= 5 0 0 0 × 3 3 7 / 4 7 = 約 3 . 6 万

40

【 0 0 6 5 】

この場合、値番号 / 値リスト 2 4 1 2 のデータ構造の転送コストの方が値配列 2 4 2 のデータ構造の転送コストよりも転送コストが小さい。そのため、結果セットデータ構造判定部 2 5 は、図 1 0 で示すように、値番号 / 値リスト 2 4 1 2 のデータ構造を選択して、値番号の再計算 (新値番号の算出) を行う。

【 0 0 6 6 】

そして、結果セットデータ構造判定部 2 5 による再計算の終了後、結果セット取得部 2 6 は、図 1 1 で示すように、値リストにヘッダ情報を付加してクライアントドライバ 1 2 に送信する。つまり、結果セット取得部 2 6 は、転送するデータが値リストであるという値リスト識別子及び参照元の表の値リストが一意に識別できる値リスト ID をヘッダ情報

50

として付加してクライアントドライバ12に値リストを転送する。

【0067】

その後、クライアントドライバ12のデータ読取部121は、受け取ったデータのヘッダ情報から都道府県の列の値リストが転送されたことを確認する。そして、データ読取部121は、都道府県の列の値リストを結果セット格納領域1221に格納する。

【0068】

上記のようにして値リストの転送が終わった後、実際にデータを取得するフェーズに入る。つまり、クライアントドライバ12は、任意の数であるnの値を動かしながら、n件目のデータをデータベースサーバ2から取得する。言い換えると、結果セット取得部26は、クライアントドライバ12からの要求に応じて、値配列242又は値番号/値リスト2412の値番号の要素の転送を行う。

10

【0069】

例えば、クライアントドライバ12は、データベースサーバ2に対して、結果セットのn件目からm件分のデータを要求したとする。

【0070】

このとき、値配列242を選択した列に対しては、結果セット取得部26は、値配列242のn番目からn+m-1番目までの配列データをクライアントドライバ12に送信する。また、この際、結果セット取得部26は、図12で示すように、値配列であることを示す識別子である値配列識別子を含んだヘッダ情報を付加して、クライアントドライバ12に対して、値配列242のn番目からn+m-1番目までの配列データを転送する。

20

【0071】

一方、値番号/値リスト2412を選択した列の場合、既に値リストは転送済みである。そのため、結果セット取得部26は、値番号のn番目からn+m-1番目までの要素をクライアントドライバ12に転送する。また、この際、結果セット取得部26は、図13で示すように、値番号であることを示す識別子である値番号識別子と、参照している値リストのID(値リストを送信する際にヘッダに付加した値リストID)と、を含んだヘッダ情報を付加して、クライアントドライバ12に対して、値番号のn番目からn+m-1番目までの要素を転送する。

【0072】

データ読取部121は、受け取ったデータのヘッダ情報を参照して、値番号と値配列242のどちらが転送されたのかを判断する。そして、データ読取部121は、受け取ったデータを結果セット格納領域1221に格納する。

30

【0073】

このような一連の処理において、クライアントドライバ12は、nもしくはmの値を変えることで、必要な行番号のデータを取得することになる。

【0074】

図8に示す例の場合では、データ読取部121は、受け取ったデータのヘッダ情報から、顧客IDの列のデータ構造が値配列242であることを識別することが出来る。また、データ読取部121は、データのヘッダ情報から、都道府県の列のデータ構造が値番号/値リスト2412であることを識別することが出来る。

40

【0075】

また、値配列242のデータ構造を有する顧客IDの列にアクセスする際には、図14に示すように、データ読取部121は、受け取った値配列242にそのままアクセスすればよいことになる。また、値番号/値リスト2412のデータ構造を有する都道府県の列にアクセスする際には、データ読取部121は、受け取った値番号が利用している値リストIDと同じIDをもつ値リストを結果セット格納領域1221から取得する。そして、データ読取部121は、n番目からn+m-1番目だけ受け取った値番号(新値番号)を値リストの添え字として、図15で示すように、値リストにアクセスすることができる。

【0076】

次に、クエリの終了について説明する。

50

【 0 0 7 7 】

クライアントドライバ 1 2 からクエリの終了を指示されると、データベースサーバ 2 の結果セット取得部 2 6 は、再計算された値番号（新値番号）、値配列 2 4 2、実行結果の表を削除する。また、クライアント 1 のクライアントドライバ 1 2 は、結果セット格納領域 1 2 2 1 に格納された値番号、値リスト、値配列 2 4 2 を削除する。

【 0 0 7 8 】

このように、クエリの終了とともに、クエリの実行により作成された各種データは削除されることになる。

【 0 0 7 9 】

次に、図 1 6 乃至図 1 8 を参照して、データベースシステム 3 の動作について説明する。

10

【 0 0 8 0 】

まず、図 1 6 を参照して、クエリの実行を行う際のデータベースシステム 3 の動作について説明する。

【 0 0 8 1 】

図 1 6 を参照すると、クライアントドライバ 1 2 は、アプリケーション 1 1 からの要求に応じて、データベースサーバ 2 に対してクエリを送信する（ステップ S 1 0 1）。

【 0 0 8 2 】

クライアントドライバ 1 2 が送信したクエリをクエリ解析部 2 1 が受信する。すると、クエリ解析部 2 1 は、受信したクエリの構文解析を実行する（ステップ S 2 0 1）。その後、クエリ解析部 2 1 は、解析結果を実行計画部 2 2 へ送信する。

20

【 0 0 8 3 】

続いて、実行計画部 2 2 は、クエリ解析部 2 1 が構文解析したクエリをどのような順序、方法で行えば効率的であるかを判定し、その実行計画を作成する（ステップ S 2 0 2）。そして、実行計画部 2 2 は、作成した実行計画をクエリ実行部 2 3 に送信する。

【 0 0 8 4 】

その後、クエリ実行部 2 3 は、実行計画部 2 2 で作成した実行計画に従ってクエリを実行する（ステップ S 2 0 3）。これにより、クエリ実行部 2 3 は、クエリに対する実行結果の表を作成して、当該作成した実行結果の表をデータ領域 2 4 の表データ 2 4 1 に格納する（ステップ S 2 0 4）。なお、クエリ実行部 2 3 が作成する実行結果の表は、F A S T 構造である。

30

【 0 0 8 5 】

データベースシステム 3 は、例えば上記動作によりクエリの実行を行う。次に、図 1 7 を参照して、結果セットの取得を行う際のデータベースシステム 3 の動作について説明する。

【 0 0 8 6 】

図 1 7 を参照すると、クライアントドライバ 1 2 は、アプリケーション 1 1 からの要求に応じて、データベースサーバ 2 の結果セット取得部 2 6 に対して結果セット取得要求を送信する（ステップ S 1 0 2）。

【 0 0 8 7 】

クライアントドライバ 1 2 から結果セット取得要求を受信すると、結果セット取得部 2 6 は、データ領域 2 4 を参照してクライアントドライバ 1 2 から受け付けたクエリに対する実行結果の表を取得する（ステップ S 3 0 1）。

40

【 0 0 8 8 】

続いて、結果セットデータ構造判定部 2 5 は、結果セット取得部 2 6 が取得した実行結果の表の各列に対して、値番号 / 値リスト 2 4 1 2 の転送コストと、値配列 2 4 2 の転送コストと、の算出を行う。そして、結果セットデータ構造判定部 2 5 は、転送コストが小さいデータ構造を選択する（ステップ S 4 0 1）。その後、結果セットデータ構造判定部 2 5 は、選択したデータ構造に応じた処理を行う（ステップ S 4 0 2）。

【 0 0 8 9 】

50

具体的には、結果セットデータ構造判定部 25 は、値番号 / 値リスト 2412 を選択した場合、値番号の添え字 (i d x) を行番号に対応させる処理を行う。つまり、結果セットデータ構造判定部 25 は、図 4 で示すように、順序集合 2411 の要素を添え字としたときの値番号の要素を、新しい値番号 (新値番号) の要素となるように、再計算する。また、値配列 242 を選択した場合、結果セットデータ構造判定部 25 は、対応する値番号 / 値リスト 2412 に基づいて、値配列 242 を作成する。つまり、結果セットデータ構造判定部 25 は、図 5 で示すように、行番号が添え字に対応するように値リストの各要素を追加することで、値配列 242 を作成する。

【 0090 】

そして、結果セット取得部 26 は、値番号 / 値リスト 2412 として選択された列の値リストをクライアントドライバ 12 に送信する (ステップ S 302)。この際、結果セット取得部 26 は、転送するデータが値リストであるという値リスト識別子及び参照元の表の値リストが一意に識別できる値リスト ID をヘッダ情報として付加してクライアントドライバ 12 に値リストを転送する。

10

【 0091 】

その後、クライアントドライバ 12 のデータ読取部 121 は、データベースサーバ 2 から転送されたデータのヘッダ情報を読み取り、値リストと識別したものをデータ領域 122 の結果セット格納領域 1221 に格納する (ステップ S 103)。

【 0092 】

上記動作により値リストが転送されると、クライアントドライバ 12 は、データベースサーバ 2 に対して、結果セットの n 件目から m 件分のデータを要求する (ステップ S 104)。つまり、クライアントドライバ 12 は、実際のデータの取得を要求する。

20

【 0093 】

クライアントドライバ 12 からデータの要求を受けると、結果セット取得部 26 は、当該要求に応じたデータをデータ領域 24 から取得して (ステップ S 303)、転送する (ステップ S 304)。

【 0094 】

結果セット取得部 26 が行うこの動作は、対象の列のデータ構造に応じて行われることになる。つまり、結果セット取得部 26 は、値配列 242 を選択した列に対して、値配列 242 の n 番目から n + m - 1 番目までの配列データをクライアントドライバ 12 に送信する。一方、値番号 / 値リスト 2412 を選択した列の場合、既に値リストは転送済みであるため、結果セット取得部 26 は、値番号の n 番目から n + m - 1 番目までの要素をクライアントドライバ 12 に転送する。

30

【 0095 】

その後、データ読取部 121 は、受け取ったデータのヘッダ情報を参照して、値番号と値配列 242 のどちらが転送されたのかを判断する。そして、データ読取部 121 は、受け取ったデータを結果セット格納領域 1221 に格納する (ステップ S 105)。

【 0096 】

データベースシステム 3 は、例えば上記動作により結果セットの取得を行う。次に、図 18 を参照して、クエリを終了する際のデータベースシステム 3 の動作について説明する。

40

【 0097 】

図 18 を参照すると、クライアントドライバ 12 は、アプリケーション 11 からの要求に応じて、クエリの終了の指示をデータベースサーバ 2 の結果セット取得部 26 に対して送信する (ステップ S 106)。

【 0098 】

上記指示を受信すると、結果セット取得部 26 は、データ領域 24 に格納された、再計算された値番号である新値番号と、値配列 242 と、を削除する (ステップ S 305)。また、結果セット取得部 26 は、データ領域 24 から実行結果の表 (表データ 241) を削除する (ステップ S 306)。

50

【 0 0 9 9 】

また、クライアントドライバ 1 2 は、データ領域 1 2 2 の結果セット格納領域 1 2 2 1 に格納された値番号、値リスト、値配列を削除する（ステップ S 1 0 7）。

【 0 1 0 0 】

データベースシステム 3 は、例えば上記動作によりクエリを終了する。

【 0 1 0 1 】

このように、本実施形態におけるデータベースサーバ 2 は、結果セットデータ構造判定部 2 5 を有している。このような構成により、結果セットデータ構造判定部 2 5 は、結果セット（クエリ実行部 2 3 により実行されたクエリの実行結果の表）の各列に対して、値番号 / 値リスト 2 4 1 2 のデータ構造を用いた場合と値配列 2 4 2 のデータ構造を用いた場合との転送コストを算出することが出来る。また、結果セットデータ構造判定部 2 5 は、転送コストの算出結果に基づいて、列ごとに転送する際のデータ構造を選択することが出来る。その結果、データベースサーバ 2 は、列ごとにデータ転送量の少ないデータ構造を選択してデータを転送することが可能となる。これにより、データベースサーバ 2 は、サーバ・クライアント間のネットワークトラフィックの増大を抑制することが可能となる。

10

【 0 1 0 2 】

[第 2 の実施形態]

本発明の第 2 の実施形態では、接続単位で値リストを再利用するデータベースシステム 5 について説明する。図 1 9 で示すように、1 回の接続で複数の問合せがある場合などにおいて、異なる問い合わせで同一の値リストを参照することがある。例えば、図 1 9 の場合では、問合せ 1 と問合せ 2 とで、どちらも同一の表の都道府県の列の値リストを参照している。このような場合には、クライアントに対象となる値リストを記憶しておくことで、後の問合せに対する結果を送信する際に再度データベースサーバ 2 から値リストを転送する必要がなくなるものと考えられる。そこで、本実施形態においては、重複して値リストを送信することを防ぐことが出来るよう構成された、データベースシステム 5 について説明する。なお、以下においては、第 1 の実施形態で説明した構成と同様の構成については、同じ符号を付すものとする。

20

【 0 1 0 3 】

図 2 0 を参照すると、本実施形態におけるデータベースシステム 5 は、クライアント 4 と、データベースサーバ 2 と、を有している。本実施形態におけるデータベースシステム 5 は、第 1 の実施形態で説明したデータベースシステム 3 とほぼ同様の構成を有している。

30

【 0 1 0 4 】

つまり、データベースサーバ 2 は、クエリ解析部 2 1 と実行計画部 2 2 と、クエリ実行部 2 3 と、データ領域 2 4 と、結果セットデータ構造判定部 2 5 と、結果セット取得部 2 6 と、を有している。また、クライアント 4 は、アプリケーション 1 1 とデータ読取部 1 2 1 と、データ領域 1 2 2 とを有しており、データ領域 1 2 2 は、結果セット格納領域 1 2 2 1 を有している。

【 0 1 0 5 】

さらに、本実施形態におけるクライアント 4 は、値リスト一時記憶領域 4 2 2 2 を有している。以下においては、データベースシステム 5 の特徴的な構成について説明する。

40

【 0 1 0 6 】

値リスト一時記憶領域 4 2 2 2 は、データベースサーバ 2 の結果セット取得部 2 6 から転送された値リストのうち、データベースサーバ 2 が保有する既存の表から参照されているものを格納する領域である。つまり、値リスト一時記憶領域 4 2 2 2 には、データベースサーバ 2 から転送された値リストのうち、クエリに応じた処理の途中で生成された値リストでなく既存の表を参照している値リストが格納されることになる。

【 0 1 0 7 】

値リスト一時記憶領域 4 2 2 2 に格納された内容は、結果セット格納領域 1 2 2 1 のデ

50

ータが破棄されても影響を受けないよう構成されている。つまり、値リスト一時記憶領域 4 2 2 2 に格納された内容は、クエリが終了するごとに破棄されないことになる。値リスト一時記憶領域 4 2 2 2 に格納された内容は、例えば、クライアント 4 とデータベースサーバ 2 との接続が切れた時に、クライアントドライバ 1 2 により削除される。

【 0 1 0 8 】

なお、値リスト一時記憶領域 4 2 2 2 が値リストを格納するか否か（値リストが既存の表から参照されているものであるか否か）は、例えば、結果セット取得部 2 6 が値リストを送信する際に付加するヘッダ情報を参照して、クライアントドライバ 1 2 により確認されることになる。

【 0 1 0 9 】

本実施形態におけるクライアントドライバ 1 2 は、結果セット取得部 2 6 から対象の列の値リストが既に送信されているか否かの確認を受け取る。例えば、クライアントドライバ 1 2 は、結果セット取得部 2 6 から対象の列の値リストの値リスト ID を受け取る。すると、クライアントドライバ 1 2 は、値リスト一時記憶領域 4 2 2 2 を参照して、対象の列の値リストが値リスト一時記憶領域 4 2 2 2 に記憶されているか否かを確認する。つまり、クライアントドライバ 1 2 は、結果セット取得部 2 6 から取得した値リスト ID と等しい ID をもつ値リストが値リスト一時記憶領域 4 2 2 2 に格納されているかを調べる。そして、クライアントドライバ 1 2 は、確認結果を結果セット取得部 2 6 に送信する。

【 0 1 1 0 】

また、クライアントドライバ 1 2 は、結果セット取得部 2 6 から値リストを受け取った場合、値リストのヘッダ情報を参照して当該受け取った値リストを結果セット格納領域 1 2 2 1 に格納するか、値リスト一時記憶領域 4 2 2 2 に格納するかを決定する。例えば、クライアントドライバ 1 2 は、値リストが既存の表を参照しているものの場合、当該値リストを値リスト一時記憶領域 4 2 2 2 に格納する。一方、クライアントドライバ 1 2 は、値リストが処理の途中で生成されたものの場合、当該値リストを結果セット格納領域 1 2 2 1 に格納する。

【 0 1 1 1 】

本実施形態における結果セットデータ構造判定部 2 5 は、クライアントドライバ 1 2 に既に転送されている値リストを加味した上で、結果セットの各列に対して値番号 / 値リスト 2 4 1 2 と値配列 2 4 2 を用いた場合の転送コストをそれぞれ算出する。そして、結果セットデータ構造判定部 2 5 は、結果セットの各列に対してどのデータ構造で転送するかを判定する。

【 0 1 1 2 】

例えば、結果セットデータ構造判定部 2 5 は、クライアントドライバ 1 2 から受け取った値リスト ID の確認結果により対象の列の値リストが既にクライアントドライバ 1 2 に転送されていると判断した場合、値番号 / 値リスト 2 4 1 2 のデータ構造を選択する。そして、結果セットデータ構造判定部 2 5 は、値番号の添え字 (i d x) を行番号に対応させる処理を行う。

【 0 1 1 3 】

なお、この場合、対象の値リストは既にクライアントドライバ 1 2 の値リスト一時記憶領域 4 2 2 2 に格納されている。そのため、結果セット取得部 2 6 による値リストの転送は省略されることになる。

【 0 1 1 4 】

また、例えば、結果セットデータ構造判定部 2 5 は、対象の列の値リストがクライアントドライバ 1 2 に転送されていないと判断した場合、値番号 / 値リスト 2 4 1 2 のデータ構造と値配列 2 4 2 のデータ構造との転送コストを算出する。そして、算出した転送コストに基づいて、転送コストの少ない方のデータ構造を選択する。

【 0 1 1 5 】

なお、本実施形態における結果セットデータ構造判定部 2 5 は、値番号 / 値リスト 2 4 1 2 の転送コストを算出する際に、対象の列に対して単一の接続でのアクセス回数が増え

10

20

30

40

50

れば増えるほどコストが下がるように転送コストを算出することが出来る。例えば、結果セットデータ構造判定部 2 5 は、下記式を用いて値番号 / 値リスト 2 4 1 2 の転送コストを算出する。

【値番号 / 値リストのデータ構造の転送コスト】

値番号の型のバイト数 × 順序集合の要素数 + 値リストの総バイト数 / (1 + 値リストのアクセス数)

なお、結果セットデータ構造判定部 2 5 は、値配列 2 4 2 の転送コストを算出する際は、例えば、第 1 の実施形態で説明した式と同じ式を用いて転送コストを算出する。

【0 1 1 6】

その後、結果セットデータ構造判定部 2 5 は、選択したデータ構造に応じた処理を行う。

10

【0 1 1 7】

例えば、結果セットデータ構造判定部 2 5 は、値番号 / 値リスト 2 4 1 2 を選択した場合、値番号の添え字 (i d x) を行番号に対応させる処理を行う。また、結果セットデータ構造判定部 2 5 は、値配列 2 4 2 を選択した場合、行番号が添え字に対応するように値リストの各要素を追加することで、値配列 2 4 2 を作成する。

【0 1 1 8】

結果セット取得部 2 6 は、クライアントドライバ 1 2 から結果セット取得要求を受け取ると、当該結果セット取得要求に応じた値リスト ID をクライアントドライバ 1 2 に送信する。これにより、結果セット取得部 2 6 は、対象の列の値リストがクライアントドライバ 1 2 に既に送信されているか (値リスト一時記憶領域 4 2 2 2 に記憶されているか) を確認する。

20

【0 1 1 9】

また、結果セット取得部 2 6 は、値リストをクライアントドライバ 1 2 に対して送信する際に、転送される値リストのヘッダ情報に、その値リストが既存の表を参照しているものであるか否かを確認するためのフラグを付加する。つまり、結果セット取得部 2 6 は、例えば図 2 1 で示すように、値リストのヘッダ情報に、値リストを一意に特定する ID の他に、その値リストが既存の表を参照しているものか、処理の途中で生成された値リストであるかを判別できるフラグを付加して転送を行う。例えば、結果セット取得部 2 6 は、サーバ内の既存の評価から作成された値リストである場合は「真」の既存表参照フラグを、処理の途中で生成された値リストの場合は「偽」の既存表参照フラグと、値リストのヘッダ情報に付加して転送する。

30

【0 1 2 0】

次に、図 2 2、図 2 3 を参照して、データベースシステム 5 の特徴的な動作について説明する。

【0 1 2 1】

図 2 2 を参照すると、クライアントドライバ 1 2 が送信した結果セット取得要求 (ステップ S 1 0 2) に応じて、結果セット取得部 2 6 は、実行結果の表を取得する (ステップ S 3 0 1) 。

【0 1 2 2】

続いて、結果セットデータ構造判定部 2 5 は、対象の列の値リストの値リスト ID をクライアントドライバ 1 2 に対して送信する (ステップ S 3 1 1) 。

40

【0 1 2 3】

上記値リスト ID を受け取ると、クライアントドライバ 1 2 は、値リスト一時記憶領域 4 2 2 2 を参照して、受け取った値リスト ID と等しい ID をもつ値リストが値リスト一時記憶領域 4 2 2 2 に格納されているかを確認する (ステップ S 1 1 1) 。そして、クライアントドライバ 1 2 は、確認結果を結果セット取得部 2 6 に送信する (ステップ S 1 1 2) 。

【0 1 2 4】

結果セットデータ構造判定部 2 5 では、クライアントドライバ 1 2 での確認結果に応じ

50

て、転送する際のデータ構造を選択する。そして、結果セットデータ構造判定部 25 は、選択したデータ構造に応じた処理を行う（ステップ S 4 1 1）。

【0125】

その後、結果セット取得部 26 は、転送コストの算出結果に基づいて選択された値番号 / 値リスト 2412 の値リストをクライアントドライバ 12 に対して転送する（ステップ S 3 1 2）。この際、結果セット取得部 26 は、値リストが既存の表を参照しているものであるか否かを確認するためのフラグをヘッダ情報に付加して値リストをクライアントドライバ 12 に対して転送する。

【0126】

そして、クライアントドライバ 12 は、転送された値リストを、当該値リストのヘッダ情報のフラグに応じた位置に格納する（ステップ S 1 1 3）。例えば、クライアントドライバ 12 は、値リストが既存の表を参照しているものの場合、当該値リストを値リスト一時記憶領域 4222 に格納する。一方、クライアントドライバ 12 は、値リストが処理の途中で生成されたものの場合、当該値リストを結果セット格納領域 1221 に格納する。

10

【0127】

次に、ステップ S 4 1 1 の動作について、図 23 を参照してより詳細に説明する。

【0128】

結果セットデータ構造判定部 25 は、クライアントドライバ 12 から送信された値リストの確認結果を受け取ると、当該確認結果に基づいて、対象の列の値リストが既にクライアントドライバ 12 に転送されているか否かを判断する（ステップ S 4 2 1）。

20

【0129】

そして、対象の列の値リストが既にクライアントドライバ 12 に転送（値リストが値リスト一時記憶領域 4222 に格納）されている場合（ステップ S 4 2 1、Yes）、結果セットデータ構造判定部 25 は、値番号 / 値リスト 2412 のデータ構造を選択する。そして、結果セットデータ構造判定部 25 は、値番号の添え字（idx）を行番号に対応させる、値番号の再計算処理を行う（ステップ S 4 2 2）。

【0130】

一方、対象の列の値リストがクライアントドライバ 12 に転送されていない場合（ステップ S 4 2 1、No）、結果セットデータ構造判定部 25 は、値番号 / 値リスト 2412 と値配列 242 を用いた場合の転送コストをそれぞれ算出する（ステップ S 4 2 3）。そして、結果セットデータ構造判定部 25 は、値番号 / 値リスト 2412 の転送コストと値配列 242 の転送コストとのどちらの転送コストが小さいかを判断する（ステップ S 4 2 4）。

30

【0131】

値番号 / 値リスト 2412 の転送コストの方が値配列 242 の転送コストよりも小さい場合、結果セットデータ構造判定部 25 は、データ構造として値番号 / 値リスト 2412 を選択する（ステップ S 4 2 4、値番号 / 値リスト）。そして、結果セットデータ構造判定部 25 は、値番号の添え字（idx）を行番号に対応させる、値番号の再計算処理を行う（ステップ S 4 2 5）。その後、結果セットデータ構造判定部 25 は結果セット取得部 26 に対して値リストを転送するよう指示する（ステップ S 4 2 6）。その結果、結果セット取得部 26 は、値リストが既存の表を参照しているものであるか否かを確認するためのフラグをヘッダ情報に付加して、値リストをクライアントドライバ 12 に対して転送することになる。

40

【0132】

また、値配列 242 の転送コストの方が値番号 / 値リスト 2412 の転送コストよりも小さい場合、結果セットデータ構造判定部 25 は、データ構造として値配列 242 を選択する（ステップ S 4 2 4、値配列）。そして、結果セットデータ構造判定部 25 は、行番号が添え字に対応するように値リストの各要素を追加することで、値配列 242 を作成する。

【0133】

50

このように、本実施形態におけるクライアントドライバ12は、値リスト一時記憶領域4222を有している。このような構成により、クライアントドライバ12は、結果セット格納領域1221のデータが破棄されても影響を受けない値リスト一時記憶領域4222に、必要な値リストを格納することが出来る。また、データベースサーバ2は、クライアントドライバ12の値リスト一時記憶領域4222に格納されている値リストを加味した上でデータ構造を選択することが出来る。ここで、結果セット格納領域1221に格納されたデータはクエリの終了とともに破棄される一方で、値リスト一時記憶領域4222に格納されたデータはクライアント4とデータベースサーバ2との接続が切れるまで破棄されない。そのため、データベースサーバ2は、クライアント4との接続が維持されている間、既に転送した値リストを再利用することが出来る。その結果、サーバ・クライアント間のネットワークトラフィックの増大をさらに抑制することが可能となる。

10

【0134】

また、結果セットデータ構造判定部25は、対象の列に対して単一の接続でのアクセス回数が増えれば増えるほどコストが下がるように値番号/値リスト2412の転送コストを算出する。このような仕組みをとることで、頻りにアクセスされる場合は値番号/値リスト2412が選択されやすくなることになる。その結果、サーバ・クライアント間のネットワークトラフィックの増大をさらに抑制することが期待できる。

【0135】

[第3の実施形態]

本発明の第3の実施形態では、値リスト一時記憶領域4222に格納された値リストが所定の条件を満たす場合、当該値リストをクライアントのストレージ装置に書き出すよう構成されたデータベースシステム7について説明する。つまり、本実施形態におけるデータベースシステム7では、接続単位ではなくクライアント単位で値リストを再利用可能なよう構成されている。なお、以下においては、第1の実施形態及び第2の実施形態で説明した構成と同様の構成については、同じ符号を付すものとする。

20

【0136】

図24を参照すると、本実施形態におけるデータベースシステム7は、クライアント6と、データベースサーバ2と、を有している。本実施形態におけるデータベースシステム7は、第1の実施形態で説明したデータベースシステム3、第2の実施形態で説明したデータベースシステム5と、ほぼ同様の構成を有している。

30

【0137】

つまり、データベースサーバ2は、クエリ解析部21と実行計画部22と、クエリ実行部23と、データ領域24と、結果セットデータ構造判定部25と、結果セット取得部26と、を有している。また、クライアント4は、アプリケーション11とデータ読取部121と、データ領域122とを有しており、データ領域122は、結果セット格納領域1221と値リスト一時記憶領域4222とを有している。

【0138】

さらに、本実施形態におけるクライアント4は、クライアントストレージ63を有している。また、クライアントストレージ63は、値リスト永続化表631と、値リスト永続記憶領域632と、を有している。また、クライアントドライバ12は、永続記憶判定部623を有している。以下においては、データベースシステム7の特徴的な構成について説明する。

40

【0139】

永続記憶判定部623は、同一の値リストが値リスト一時記憶領域4222に格納される回数を、値リスト永続化表631を用いて監視する。また、永続記憶判定部623は、同一の値リストが値リスト一時記憶領域4222に格納された回数と予め定められた永続化閾値との比較を行う。そして、永続記憶判定部623は、同じ値リストが値リスト一時記憶領域4222に格納された回数が永続化閾値以上になった場合、値リスト永続記憶領域632に値リストを書き出す処理を行う。

【0140】

50

また、永続記憶判定部 6 2 3 は値リスト永続記憶領域 6 3 2 に格納されている値リストが必要になった場合に、値リスト永続記憶領域 6 3 2 から値リスト一時記憶領域 4 2 2 2 に値リストをアップロードする。例えば、永続記憶判定部 6 2 3 は、結果セット取得部 2 6 から送信された値リストの ID に応じて、当該値リスト ID に応じた値リストが値リスト永続記憶領域 6 3 2 に記憶されているか否かを判断する。そして、永続記憶判定部 6 2 3 は、値リストが値リスト永続記憶領域 6 3 2 に記憶されている場合、値リスト永続記憶領域 6 3 2 から値リスト一時記憶領域 4 2 2 2 に値リストをアップロードする。

【 0 1 4 1 】

クライアントドライバ 1 2 は、値リスト永続記憶領域 6 2 3 に記憶されている値リストも加味して、対象の列の値リストが既に送信されているか否かの確認を行うことになる。

【 0 1 4 2 】

クライアントストレージ 6 3 は、ディスク装置などの記憶装置である。クライアントストレージ 6 3 は、値リスト永続化表 6 3 1 と、値リスト永続記憶領域 6 3 2 と、を有している。

【 0 1 4 3 】

値リスト永続化表 6 3 1 は、同一の値リストが値リスト一時記憶領域 4 2 2 2 に格納された回数を数える際に、永続記憶判定部 6 2 3 により利用される。

【 0 1 4 4 】

図 2 5 は、値リスト永続化表 6 3 1 の構成の一例である。図 2 5 を参照すると、値リスト永続化表 6 3 1 は、例えば、値リスト ID と参照回数と、永続化フラグと、値リストの格納場所と、を有している。

【 0 1 4 5 】

値リスト ID は、値リストの ID を示している。永続記憶判定部 6 2 3 は、値リスト ID を利用して、値リスト永続化表の参照回数を更新することになる。

【 0 1 4 6 】

参照回数は、異なる接続で同一の値リストが値リスト一時記憶領域 4 2 2 2 に格納されるごとに、永続記憶判定部 6 2 3 により 1 加算される。従って、永続記憶判定部 6 2 3 は、例えば、結果セット取得部 2 6 から転送された値リストが値リスト一時記憶領域 4 2 2 2 に格納されるごとに、参照回数を 1 加算する。また、永続記憶判定部 6 2 3 は、値リスト永続記憶領域 6 3 2 から値リストを値リスト一時記憶領域 4 2 2 2 にアップロードするごとに、参照回数を 1 加算する。例えば、図 2 5 の 1 行目 (値リスト ID 「 A F 6 C D 5 4 2 ... 」) は、対象の値リストが値リスト一時記憶領域 4 2 2 2 に格納された回数が 1 回であることを示している。

【 0 1 4 7 】

永続化フラグは、同じ値リストが値リスト一時記憶領域 4 2 2 2 に格納された回数が永続化閾値以上になった場合に真となるフラグである。上述したように、永続化フラグは、永続記憶判定部 6 2 3 により操作される。例えば、図 2 5 では、値リスト ID 「 9 7 2 B C 7 3 5 ... 」の永続化フラグが真であり、値リスト ID 「 D 2 6 9 0 B 4 3 ... 」の永続化フラグが偽であることを示している。このように、図 2 5 では、永続化閾値が 3 である場合を示している。なお、永続化閾値は任意の数で構わない。永続化閾値は、予め定められていても構わないし、ユーザにより選択されるように構成しても構わない。

【 0 1 4 8 】

値リストの格納場所は、値リスト永続記憶領域 6 3 2 のうちの対象の値リストが書き込まれた位置を示している。上記のように、永続記憶判定部 6 2 3 は、永続化フラグが真になった値リストを値リスト永続記憶領域 6 3 2 に書き出すとともに、その格納パスを値リスト永続化表 6 3 1 に記録する。そのため、永続化フラグが偽である値リストの格納場所は、空欄になっている。

【 0 1 4 9 】

値リスト永続記憶領域 6 3 2 には、永続記憶判定部 6 2 3 により、値リスト永続化表 6 3 1 の永続化フラグが真になった値リストが格納される。値リスト永続記憶領域 6 3 2 に

10

20

30

40

50

格納された値リストは、必要に応じて、永続記憶判定部 6 2 3 により値リスト一時記憶領域 4 2 2 2 に書き出される。

【 0 1 5 0 】

このように、本実施形態におけるクライアント 6 は、永続記憶判定部 6 2 3 と値リスト永続化表 6 3 1 と値リスト永続記憶領域 6 3 2 とを有している。このような構成により、クライアント 6 の永続記憶判定部 6 2 3 は、値リスト一時記憶領域 4 2 2 2 に格納された値リストの格納回数を、値リスト永続化表 6 3 1 を利用して数えることが出来る。また、値リストの格納回数により当該値リストの利用回数が多いと判断される場合、永続記憶判定部 6 2 3 は、値リスト一時記憶領域 4 2 2 2 に格納された値リストを値リスト永続記憶領域 6 3 2 に書き出すことが出来る。その結果、値リスト永続記憶領域 6 3 2 に書き出された値リストは、例えば、クエリが終了しても、クライアント 6 とデータベースサーバ 2 との接続が終了しても削除されず、クライアント 6 に記憶され続けることになる。これにより、同一クライアントであれば異なる接続であっても値リストを共有することが可能となり、その結果、サーバ・クライアント間のネットワークトラフィックの増大をさらに抑制することが可能となる。

10

【 0 1 5 1 】

[第 4 の実施形態]

本発明の第 4 の実施形態では、データベースサーバ 8 2 に対して問合せを実行するクライアント 8 1 と、クライアント 8 1 からの問合せに応じて、各問合せに対する回答をレコードで表す問合せの実行結果の表を送信するデータベースサーバ 8 2 と、を有するデータベースシステム 8 について説明する。なお、本実施形態では、データベースシステム 8 の構成の概要について説明する。

20

【 0 1 5 2 】

図 2 6 を参照すると、データベースシステム 8 は、クライアント 8 1 とデータベースサーバ 8 2 とを有している。

【 0 1 5 3 】

クライアント 8 1 は、データベースサーバ 8 2 に対して問合せを実行する。

【 0 1 5 4 】

データベースサーバ 8 2 は、データ送信部 8 2 1 を有している。

【 0 1 5 5 】

データ送信部 8 2 1 は、複数の列を有する実行結果の表を送信する際に、当該実行結果の表の列ごとに、当該列を表す予め定められた複数のデータ構造のうちのいずれかを用いて送信する。

30

【 0 1 5 6 】

このように、本実施形態におけるデータベースシステム 8 のデータベースサーバ 8 2 は、データ送信部 8 2 1 を有している。このような構成により、データ送信部 8 2 1 は、複数の列を有する実行結果の表を送信する際に、当該実行結果の表の列ごとに、当該列を表す予め定められた複数のデータ構造のうちのいずれかを用いて送信することが出来る。その結果、データベースサーバ 8 2 は、実行結果の表の列ごとにデータ転送量などの点から望ましいデータ構造を選択して、列ごとに実行結果の表を転送することが可能となる。これにより、データベースシステム 8 は、サーバ・クライアント間のネットワークトラフィックの増大を抑制することが可能となる。

40

【 0 1 5 7 】

また、上述したデータベースシステム 8 と同様の効果は、データベースサーバ 8 2 単独でも実現することが出来る。具体的に、本発明の他の形態であるデータベースサーバ 8 2 は、クライアントからの問合せに応じて、各問合せに対する回答をレコードで表す問合せの実行結果の表を送信するデータベースサーバ 8 2 であって、複数の列を有する実行結果の表を送信する際に、当該実行結果の表の列ごとに、当該列を表す予め定められた複数のデータ構造のうちのいずれかを用いて送信するデータ送信部を有する、という構成を有する。

50

【 0 1 5 8 】

また、上記データベースサーバ 8 2 は、当該データベースサーバ 8 2 に所定のプログラムが組み込まれることで実現できる。具体的に、本発明の他の形態であるプログラムは、クライアントからの問合せに応じて、各問合せに対する回答をレコードで表す問合せの実行結果の表を送信するデータベースサーバに、複数の列を有する前記実行結果の表を送信する際に、当該実行結果の表の列ごとに、当該列を表す予め定められた複数のデータ構造のうちのいずれかを用いて送信するデータ送信手段を実現させるためのプログラムである。

【 0 1 5 9 】

また、上述したデータベースサーバ 8 2 が作動することにより実行される情報処理方法は、クライアントからの問合せに応じて、各問合せに対する回答をレコードで表し複数の列項目を有する問合せの実行結果の表を送信する際に、実行結果の表の列ごとに、当該列を表す予め定められた複数のデータ構造のうちのいずれかを用いて送信する、という方法である。

10

【 0 1 6 0 】

上述した構成を有する、データベースサーバ、プログラム、又は、情報処理方法、の発明であっても、上記データベースシステム 8 と同様の作用を有するために、上述した本発明の目的を達成することが出来る。

【 0 1 6 1 】

< 付記 >

20

上記実施形態の一部又は全部は、以下の付記のようにも記載されうる。以下、本発明における分散処理制御装置などの概略を説明する。但し、本発明は、以下の構成に限定されない。

【 0 1 6 2 】

(付記 1)

データベースサーバに対して問合せを実行するクライアントと、前記クライアントからの問合せに応じて、各問合せに対する回答をレコードで表し、当該レコードを含む問合せの実行結果の表を送信するデータベースサーバと、を有するデータベースシステムであって、

前記データベースサーバは、複数の列を有する前記実行結果の表を送信する際に、当該実行結果の表の列ごとに、当該列を表すことが出来る複数のデータ構造のうちのいずれかを用いて送信するデータ送信部を有する

30

データベースシステム。

【 0 1 6 3 】

(付記 2)

付記 1 に記載のデータベースシステムであって、

前記データ送信部は、前記複数のデータ構造のそれぞれを用いた際のデータを前記クライアントに転送する際のデータ転送量に基づいて選択されたデータ構造を用いて、前記実行結果の表の列ごとにデータを送信する

データベースシステム。

40

【 0 1 6 4 】

(付記 3)

付記 2 に記載のデータベースシステムであって、

前記データ送信部は、前記複数のデータ構造のうちの前記データ転送量が最も小さくなるデータ構造を用いて、前記実行結果の表の列ごとにデータを送信する

データベースシステム。

【 0 1 6 5 】

(付記 4)

付記 1 乃至 3 のいずれかに記載のデータベースシステムであって、

前記実行結果の表の各列に対して、複数のデータ構造におけるデータ転送量を示す転送

50

コストをそれぞれ算出し、当該算出した転送コストに基づいてデータを転送する際のデータ構造を列ごとに選択するデータ構造選択部を有し、

前記データ送信部は、前記データ構造選択部が列ごとに選択したデータ構造を用いて、列ごとにデータを送信する
を有する

データベースシステム。

【0166】

(付記5)

付記4に記載のデータベースシステムであって、

前記データ構造選択部は、データを値の一覧である値リストと前記値リストのうちの該当する位置を示す値番号との組み合わせで表現するデータ構造と、行番号に対応する値の配列で構成されるデータ構造と、の前記転送コストを算出し、当該算出した転送コストに基づいてデータを転送する際のデータ構造を列ごとに選択する

データベースシステム。

【0167】

(付記5-1)

付記5に記載のデータベースシステムであって、

データ構造選択部は、前記値リストと前記値番号の組み合わせで表現するデータ構造を選択する場合、前記値番号の添え字を行番号に対応させる処理を行う

データベースシステム。

【0168】

(付記5-2)

付記4乃至5-1のいずれかに記載のデータベースシステムであって、

前記データ構造選択部は、前記データ構造ごとの予想バイト数を算出することで、前記転送コストを算出する

データベースシステム。

【0169】

(付記5-3)

付記4乃至5-2のいずれかに記載のデータベースシステムであって、

前記データ構造選択部は、前記データ構造ごとの予想バイト数とデータに対するアクセスの回数とに基づいて前記転送コストを算出する

データベースシステム。

【0170】

(付記6)

付記1乃至5に記載のデータベースシステムであって、

前記クライアントは、前記データベースサーバから送信されたデータを一時的に記憶する一時データ記憶部を有し、

前記データベースサーバの前記データ送信部は、前記一時データ記憶部が記憶するデータに基づいて選択された前記複数のデータ構造のうちのいずれかを用いて送信する

データベースシステム。

【0171】

(付記7)

付記5に記載のデータベースシステムであって、

前記クライアントは、前記一時データ記憶部に記憶されたデータが所定の条件を満たす場合に、当該一時データ記憶部が記憶するデータをストレージ装置に移動させる

データベースシステム。

【0172】

(付記8)

クライアントからの問合せに応じて、各問合せに対する回答をレコードで表し、当該レコードを含む問合せの実行結果の表を送信するデータベースサーバであって、

10

20

30

40

50

複数の列を有する前記実行結果の表を送信する際に、当該実行結果の表の列ごとに、当該列を表すことが出来る複数のデータ構造のうちのいずれかを用いて送信するデータ送信部を有する

データベースサーバ。

【0173】

(付記8-1)

付記8に記載のデータベースサーバであって、

前記データ送信部は、前記複数のデータ構造のそれぞれを用いた際のデータを前記クライアントに転送する際のデータ転送量に基づいて選択されたデータ構造を用いて、前記実行結果の表の列ごとにデータを送信する

データベースサーバ。

【0174】

(付記8-2)

付記8又は付記8-1に記載のデータベースサーバであって、

前記データ送信部は、前記複数のデータ構造のうちの前記データ転送量が最も小さくなるデータ構造を用いて、前記実行結果の表の列ごとにデータを送信する

データベースシステム。

【0175】

(付記9)

クライアントからの問合せに応じて、各問合せに対する回答をレコードで表し、当該レコードを含み複数の列項目を有する問合せの実行結果の表を送信する際に、

前記実行結果の表の列ごとに、当該列を表すことが出来る複数のデータ構造のうちのいずれかを用いて送信する

情報処理方法。

【0176】

(付記9-1)

付記9に記載の情報処理方法であって、

前記複数のデータ構造のそれぞれを用いた際のデータを前記クライアントに転送する際のデータ転送量に基づいて選択されたデータ構造を用いて、前記実行結果の表の列ごとにデータを送信する

情報処理方法。

【0177】

(付記9-2)

付記9又は付記9-1に記載の情報処理方法であって、

前記複数のデータ構造のうちの前記データ転送量が最も小さくなるデータ構造を用いて、前記実行結果の表の列ごとにデータを送信する

情報処理方法。

【0178】

(付記10)

クライアントからの問合せに応じて、各問合せに対する回答をレコードで表し、当該レコードを含む問合せの実行結果の表を送信するデータベースサーバに、

複数の列を有する前記実行結果の表を送信する際に、当該実行結果の表の列ごとに、当該列を表すことが出来る複数のデータ構造のうちのいずれかを用いて送信するデータ送信手段を実現させるための

プログラム。

【0179】

(付記10-1)

付記10に記載のプログラムであって、

前記データ送信手段は、前記複数のデータ構造のそれぞれを用いた際のデータを前記クライアントに転送する際のデータ転送量に基づいて選択されたデータ構造を用いて、前記

10

20

30

40

50

実行結果の表の列ごとにデータを送信するプログラム。

【0180】

(付記10-2)

付記10又は付記10-1に記載のプログラムであって、

前記データ送信部は、前記複数のデータ構造のうちの前記データ転送量が最も小さくなるデータ構造を用いて、前記実行結果の表の列ごとにデータを送信するプログラム。

【0181】

なお、上記各実施形態及び付記において記載したプログラムは、記憶装置に記憶されていたり、コンピュータが読み取り可能な記録媒体に記録されていたりする。例えば、記録媒体は、フレキシブルディスク、光ディスク、光磁気ディスク、及び、半導体メモリ等の可搬性を有する媒体である。

10

【0182】

以上、上記各実施形態を参照して本願発明を説明したが、本願発明は、上述した実施形態に限定されるものではない。本願発明の構成や詳細には、本願発明の範囲内で当業者が理解しうる様々な変更をすることが出来る。

【符号の説明】

【0183】

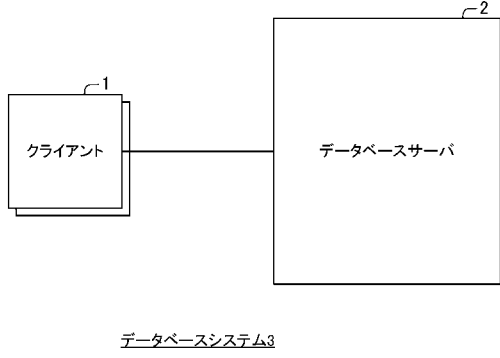
- 1、4、6 クライアント
- 11 アプリケーション
- 12 クライアントドライバ
- 121 データ読取部
- 122 データ領域
- 1221 結果セット格納領域
- 2 データベースサーバ
- 21 クエリ解析部
- 22 実行計画部
- 23 クエリ実行部
- 24 データ領域
- 241 表データ
- 2411 順序集合
- 2412 値番号/値リスト
- 242 値配列
- 25 結果セットデータ構造判定部
- 26 結果セット取得部
- 3、5、7 データベースシステム
- 4222 値リスト一時記憶領域
- 623 永続記憶判定部
- 63 クライアントストレージ
- 631 値リスト永続化表
- 632 値リスト永続記憶領域

20

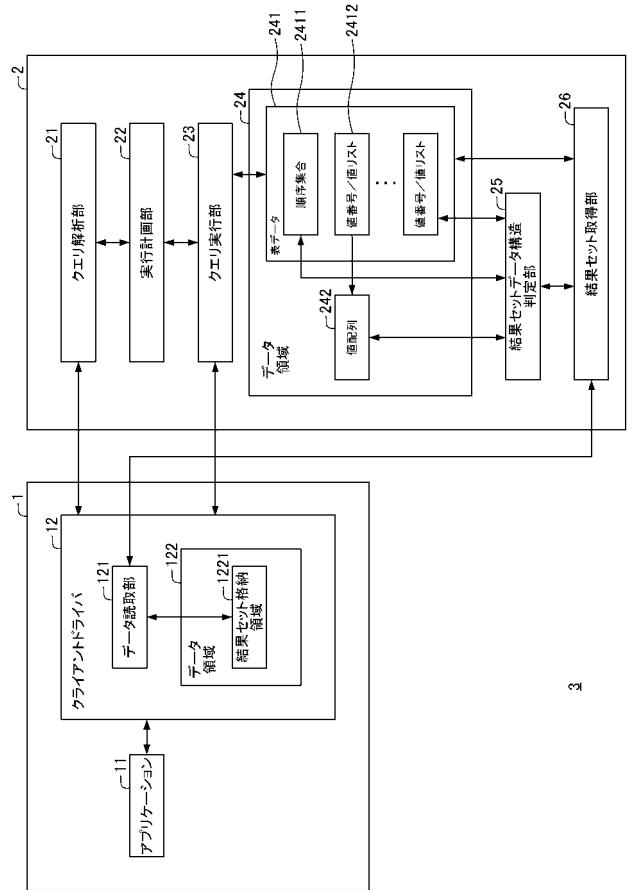
30

40

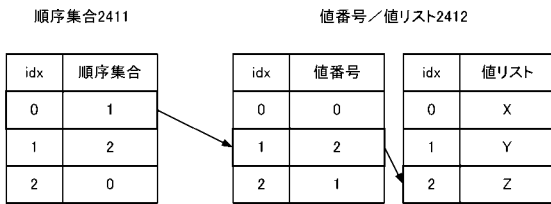
【図1】



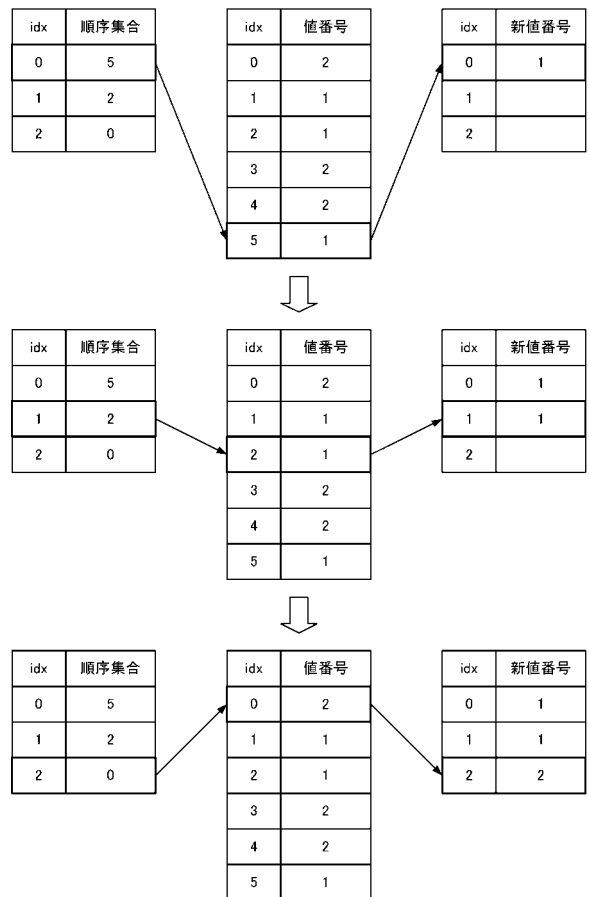
【図2】



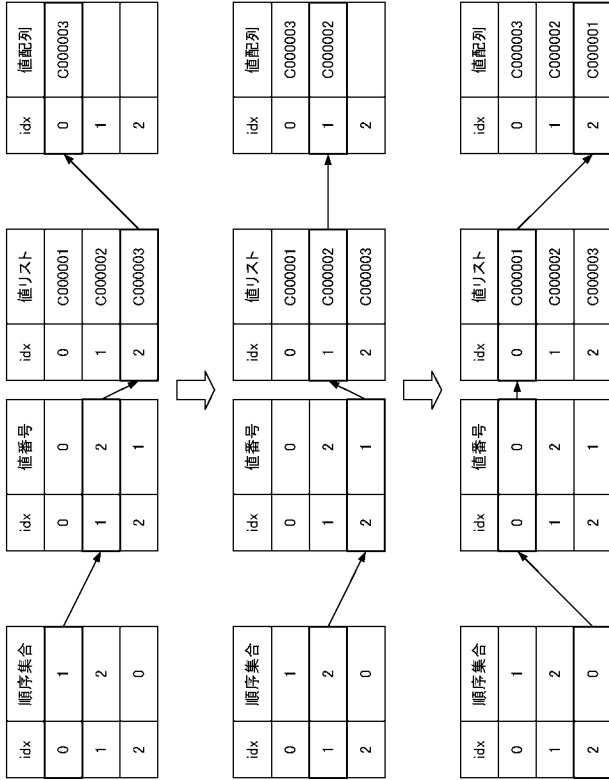
【図3】



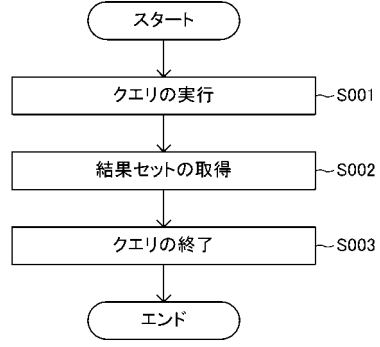
【図4】



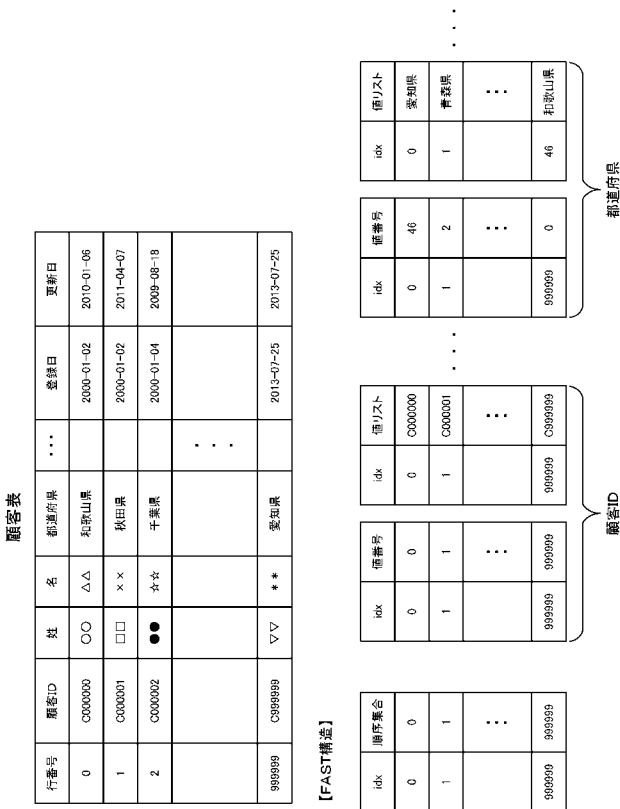
【図 5】



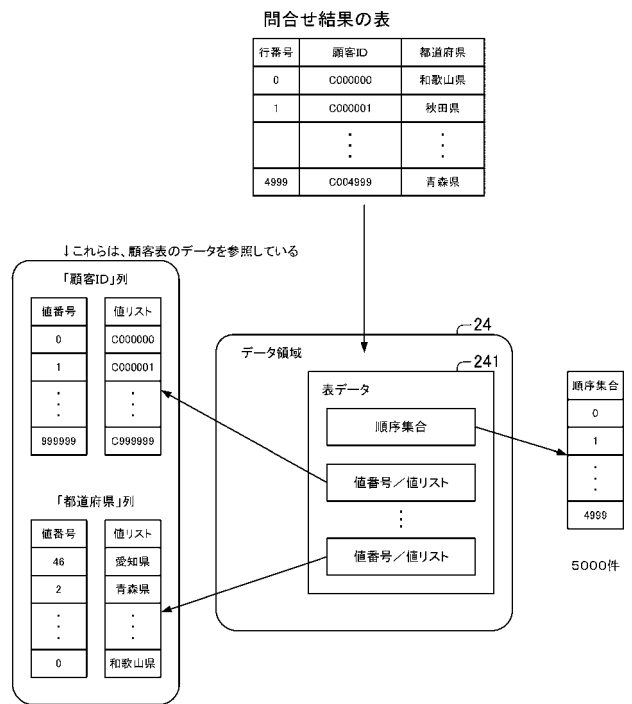
【図 6】



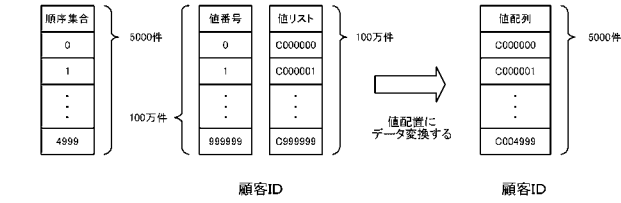
【図 7】



【図 8】



【 図 9 】

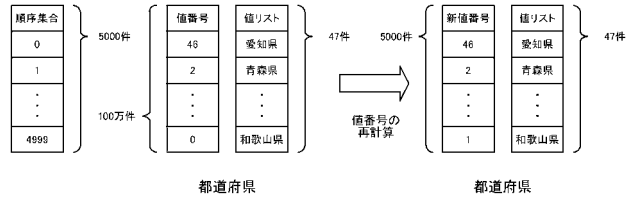


【値番号／値リストのコスト】=
 順序集合の要素数×値番号の型のバイト数+値リストの総バイト数
 =5000×4(unit型)+800万=802万

【値配列のコスト】=
 順序集合の要素数×値番号の型のバイト数+値リストの総バイト数
 =5000×800万/100万=4万

} 値配列を採用する

【 図 1 0 】

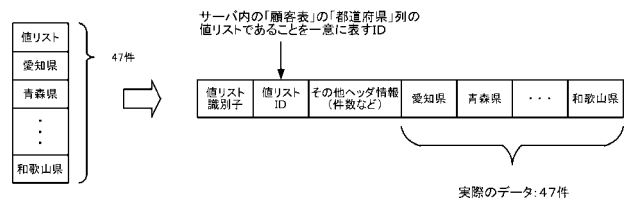


【値番号／値リストのコスト】=
 順序集合の要素数×値番号の型のバイト数+値リストの総バイト数
 =5000×1(uchar型)+337=5000

【値配列のコスト】=
 順序集合の要素数×値番号の型のバイト数+値リストの総バイト数
 =5000×337/47=3.6万

} 値番号／値リストを採用する

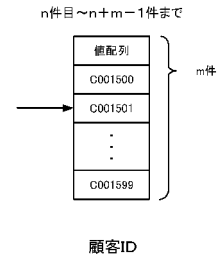
【 図 1 1 】



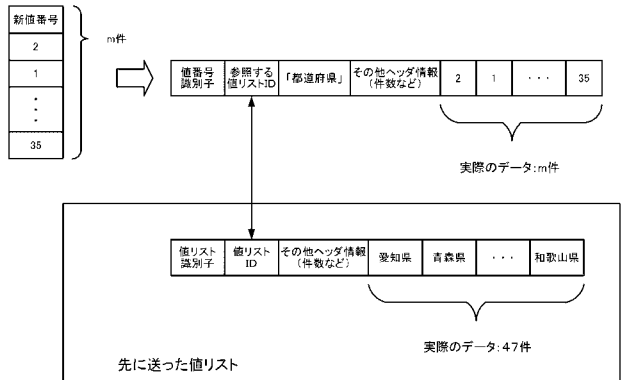
【 図 1 2 】



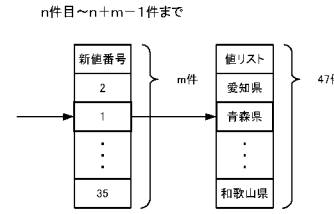
【 図 1 4 】



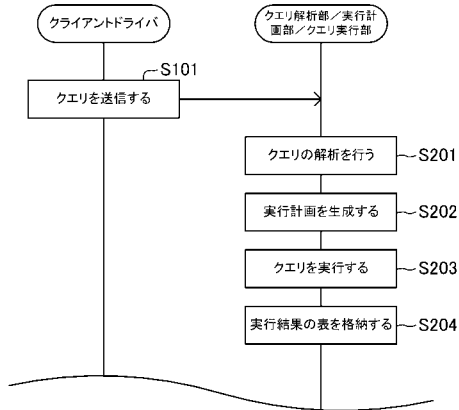
【 図 1 3 】



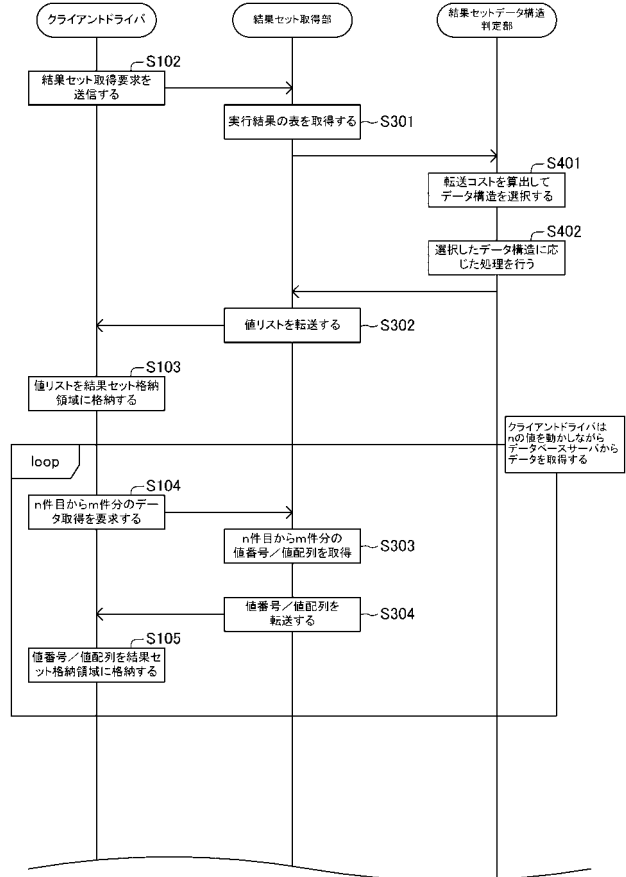
【 図 1 5 】



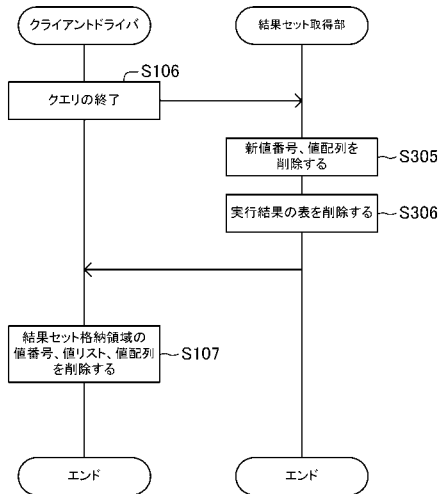
【図16】



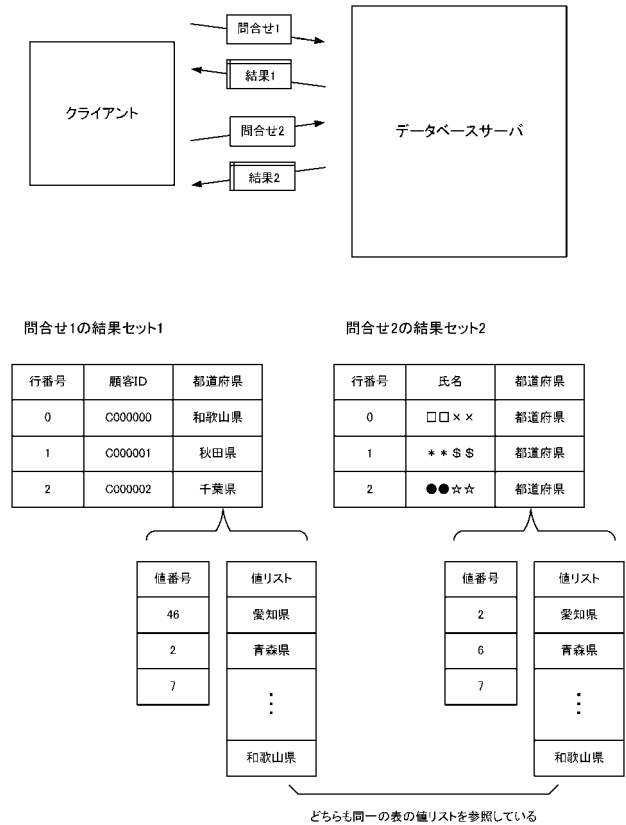
【図17】



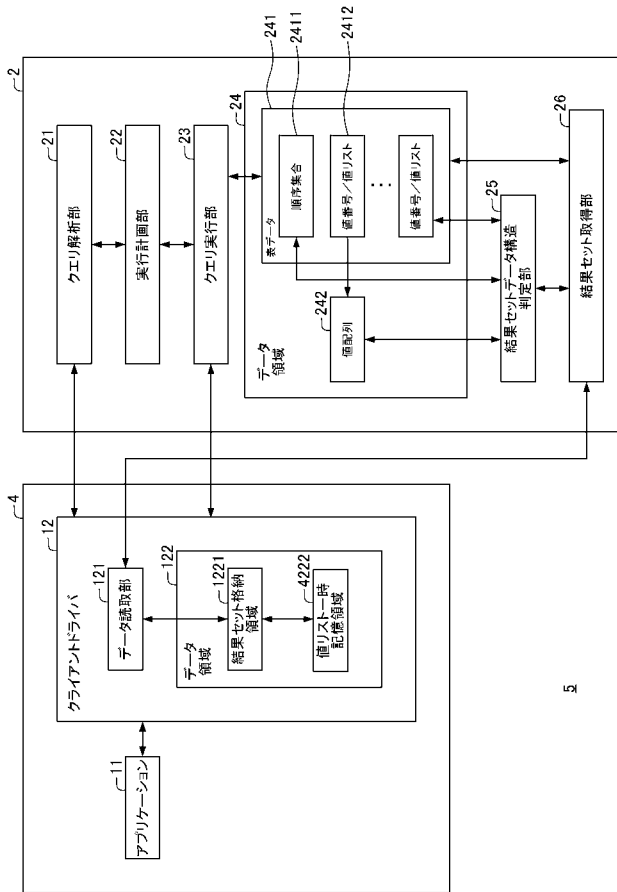
【図18】



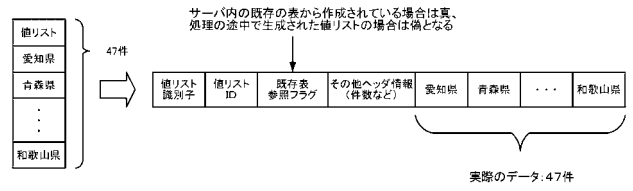
【図19】



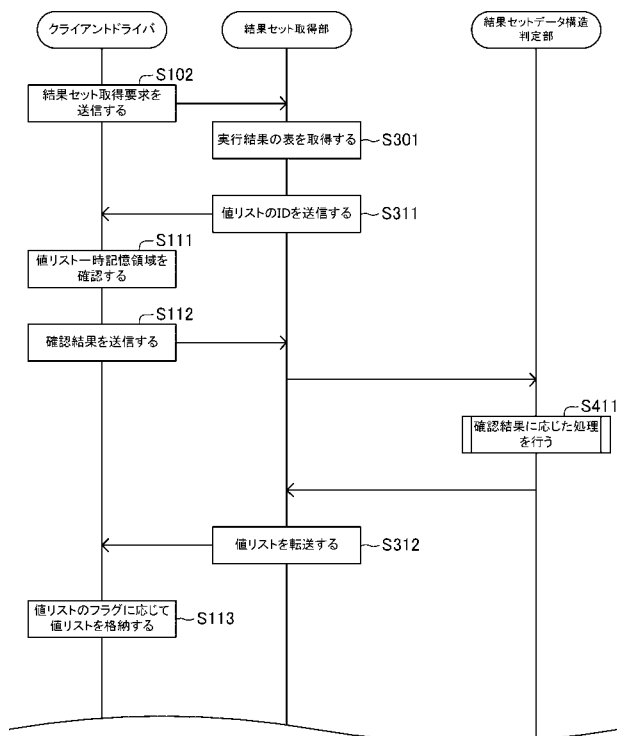
【図 20】



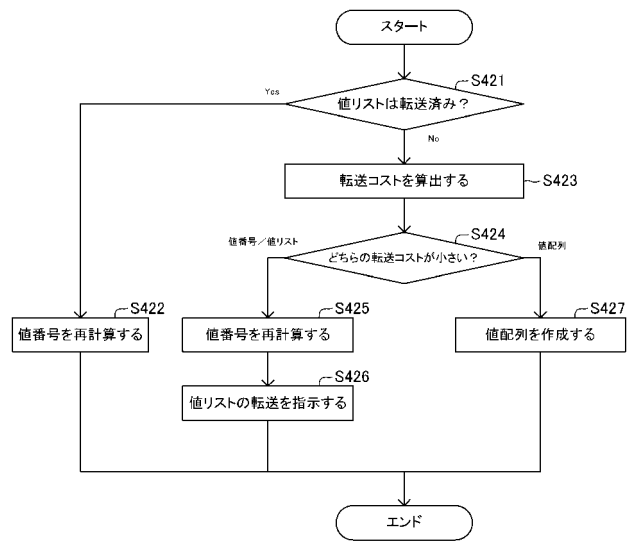
【図 21】



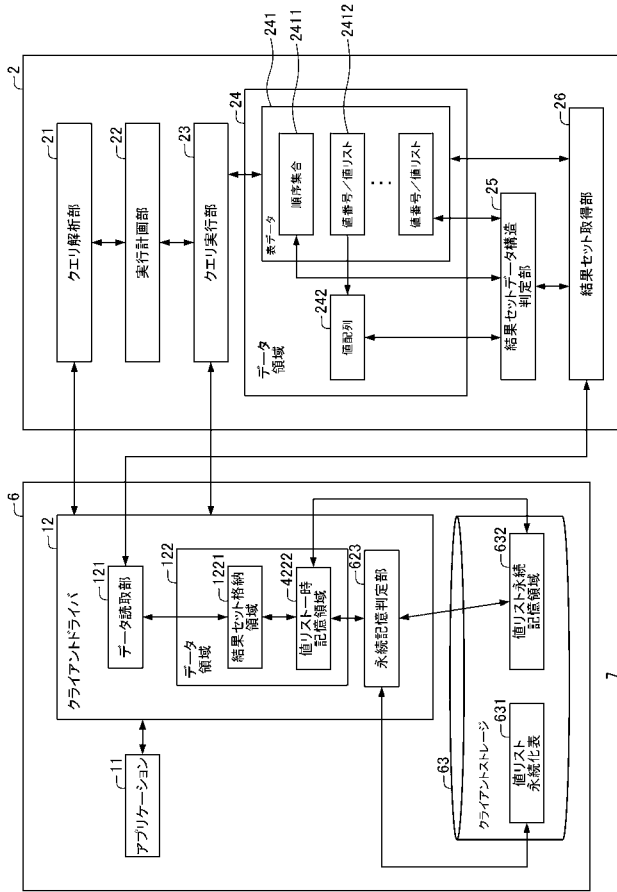
【図 22】



【図 23】



【図 24】



【図 25】

値リストID	参照回数 (異なる接続でインクリメント)	永続化フラグ	値リストの格納場所
AF6CD542...	1	偽	
302CD641...	75	真	/var/cache/xxx/302C...
972BC735...	4	真	/var/cache/xxx/972B...
...			
D2690B43...	2	偽	

【図 26】

