

(12) 特許協力条約に基づいて公開された国際出願

(19) 世界知的所有権機関
国際事務局



(43) 国際公開日
2011年10月13日(13.10.2011)

PCT

(10) 国際公開番号
WO 2011/125237 A1

- (51) 国際特許分類:
G06F 12/00 (2006.01) G06F 17/30 (2006.01)
- (21) 国際出願番号: PCT/JP2010/062343
- (22) 国際出願日: 2010年7月22日(22.07.2010)
- (25) 国際出願の言語: 日本語
- (26) 国際公開の言語: 日本語
- (30) 優先権データ:
特願 2010-090491 2010年4月9日(09.04.2010) JP
- (71) 出願人 (米国を除く全ての指定国について): 株式会社日立製作所 (HITACHI, LTD.) [JP/JP]; 〒1008280 東京都千代田区丸の内一丁目6番6号 Tokyo (JP).
- (72) 発明者; および
- (75) 発明者/出願人 (米国についてのみ): 中野 定樹 (NAKANO, Sadaki) [JP/JP]; 〒1858601 東京都国分寺市東恋ヶ窪一丁目280番地 株式会社日立製作所中央研究所内 Tokyo (JP). 室 啓朗 (MURO, Keiro) [JP/JP]; 〒1858601 東京都国分寺市

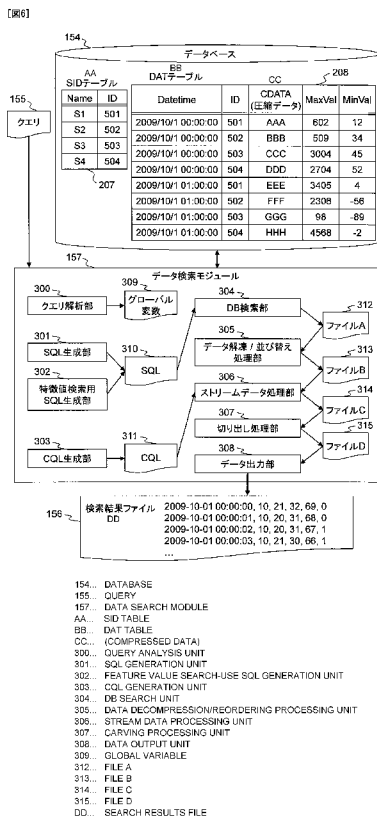
東恋ヶ窪一丁目280番地 株式会社日立製作所中央研究所内 Tokyo (JP). 藤原 真二 (FUJIWARA, Shinji) [JP/JP]; 〒1858601 東京都国分寺市東恋ヶ窪一丁目280番地 株式会社日立製作所中央研究所内 Tokyo (JP).

- (74) 代理人: 後藤 政喜 (GOTO, Masaki); 〒1000013 東京都千代田区霞が関三丁目3番1号尚友会館 Tokyo (JP).
- (81) 指定国 (表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

[続葉有]

(54) Title: DATABASE MANAGEMENT METHOD, COMPUTER, SENSOR NETWORK SYSTEM AND DATABASE SEARCH PROGRAM

(54) 発明の名称: データベース管理方法、計算機、センサネットワークシステム及びデータベース検索プログラム



(57) Abstract: Disclosed is a database management method for reducing the amount of data stored in a database and facilitating tracking for identifying the site of an anomaly and the cause of the anomaly in a system for handling a multitude of sensor information of a plant. The disclosed database management method in a computer for managing a database includes a step, upon receiving a query, for the computer to analyze the query; a step for generating a first inquiry for searching compressed data on the basis of the analysis results of the query; a step for generating a second inquiry for executing a search with respect to time-series data; a step for extracting predetermined data from a plurality of time-series data that have been acquired on the basis of response results to the second inquiry; and a step for extracting data for outputting to a client computer from the extracted predetermined data and generating an output result.

(57) 要約: プラント等の多数のセンサ情報を扱うシステムにおいて、データベースに格納されるデータ量を削減し、異常箇所及び異常原因の特定のための追跡を容易にすることを目的とする。データベースを管理する計算機におけるデータベース管理方法であって、クエリを受け付けた場合に、計算機が、クエリを解析するステップと、クエリの解析結果に基づいて圧縮データを検索するための第1の問い合わせを生成するステップと、時系列データに対する検索を実行するための第2の問い合わせを生成するステップと、第2の問い合わせに対する応答結果に基づいて、取得された複数の時系列データから所定のデータを抽出するステップと、抽出された所定のデータからクライアント計算機に出力するためのデータを抽出し、出力結果を生成するステップと、を含むことを特徴とする。



(84) 指定国 (表示のない限り、全ての種類の広域保護が可能): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), ヨーロッパ (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, SM, TR), OAPI (BF,

BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

添付公開書類:

— 国際調査報告 (条約第 21 条(3))

明 細 書

発明の名称：

データベース管理方法、計算機、センサネットワークシステム及びデータベース検索プログラム

技術分野

[0001] 本発明は、データ処理を行うための計算機システム及びそのデータ処理方法に係り、特に、工場やプラントにおける設備等の予防保全を目的にして収集される大量のデータを処理するに適した計算機システム及びそのデータ処理方法に関する。

背景技術

[0002] 火力発電プラントでは、数千にも及ぶセンサが施設の様々な部位に取り付けられている。それらセンサは、秒単位、又は、ミリ秒単位で計測結果を常時発信し続けており、プラント生涯に渡って発信されるデータ量は数百～数十テラバイトと膨大なデータ量になると試算される。これら多年度にわたって日々刻々と送られてくるデータを管理するシステムでは、データベースに格納されるデータ量が膨大となるため、データベースのディスク容量の増加が課題となる。

[0003] 一方、プラントにおいて異常が発生した場合に、プラントの管理者は、当該データベースに格納されたセンサ情報を解析して異常が発生した原因を早急に特定する必要がある。しかし、異常値を示すセンサデータ（信号）をビューア等に表示しながら、問題発生の原因を絞り込むために、データベース上の膨大なデータに対して何度も検索クエリを発行する必要があり、異常原因の特定まで時間がかかる。

[0004] これを解決する方法としては、例えば、特許文献１に記載されるように、センサ情報を圧縮してデータベースに格納することによって、ディスク容量の増加を解消することが可能である。また、異常原因の特定では、データベースに発行するクエリ数を削減することによって異常原因の特定が短時間で

行える。

先行技術文献

特許文献

[0005] 特許文献1：特開2002-358117号公報

発明の概要

発明が解決しようとする課題

[0006] しかし、前述した方法では、圧縮データを検索する場合、データを一旦何処かへ展開したのち、展開後のデータに対して検索を行う必要が生じるため検索速度が犠牲になるという問題があった。また、異常の原因を特定する場合に、センサ情報から異常箇所の検索のための検索クエリ、及び異常が発生した原因を特定するために異常箇所の追跡を行うための検索クエリを複数回発行する必要があった。したがって、発行されるクエリ数の増大と異常箇所の検索時間の増大という問題があった。

[0007] 本発明は、前述した問題点に鑑みてなされた発明である。すなわち、データベースに格納されるデータ量を削減し、大量のデータの中から所望とするデータを効率よく検索することを目的とする。

課題を解決するための手段

[0008] 本発明の代表的な一例を示せば以下の通りである。すなわち、プロセッサと、前記プロセッサに接続されるメモリとを備え、データベースを管理する計算機におけるデータベース管理方法であって、前記データベースは、所定の条件に基づいて圧縮された複数の圧縮データを格納し、前記データベース管理方法は、前記計算機に接続されるクライアント計算機から前記データベースへのクエリを受け付けた場合に、前記計算機が、前記受け付けたクエリを解析する第1のステップと、前記計算機が、前記受け付けたクエリの解析結果に基づいて、前記データベースから一つ以上の前記圧縮データを検索するための第1の問い合わせを生成する第2のステップと、前記計算機が、前記受け付けたクエリの解析結果に基づいて、前記第1の問い合わせの応答結

果である前記一つ以上の圧縮データから取得される複数の時系列データに対する検索を実行するための第2の問い合わせを生成する第3のステップと、前記計算機が、前記データベースに前記第1の問い合わせを発行して、前記データベースから前記第1の問い合わせの応答結果として一つ以上の前記圧縮データを取得する第4のステップと、前記計算機が、前記第1の問い合わせに対する応答結果として取得された一つ以上の圧縮データを解凍することによって前記複数の時系列データを取得する第5のステップと、前記計算機が、前記取得された複数の時系列データに対して前記第2の問い合わせを実行する第6のステップと、前記計算機が、前記第2の問い合わせに対する応答結果に基づいて、前記取得された複数の時系列データから所定のデータを抽出する第7のステップと、前記計算機が、前記第7のステップにおいて抽出された所定のデータから前記クライアント計算機に出力するためのデータを抽出し、出力結果を生成する第8のステップと、を含むことを特徴とする。

発明の効果

[0009] 本発明によれば、実質的なデータを維持しつつデータベースに格納されるデータ量を削減し、また、大量なデータから所望とするデータを効率よく検索することができる。

図面の簡単な説明

[0010] [図1]本発明が適用されたセンサネットワークシステムの一実施形態における計算機システムの概要を説明するブロック図である。

[図2]本実施形態におけるセンタサーバのハードウェア構成を説明するブロック図である。

[図3]本実施形態におけるセンタネットワークシステムのソフトウェア構成を説明するブロック図である。

[図4]本実施形態におけるデータローダモジュール及びデータベースを説明するブロック図である。

[図5A]本実施形態において、データローダモジュールによって実行される処

理を説明するフローチャートである。

[図5B]本実施形態において、データローダモジュールによって実行されるデータロード処理を説明するフローチャートである。

[図5C]本実施形態において、データローダモジュールによって実行されるデータロード処理を説明するフローチャートである。

[図6]本実施形態におけるデータ検索モジュールを説明するブロック図である。

[図7]本実施形態において、データ検索モジュールによって出力されるデータの構造の一例を示す説明図である。

[図8A]本実施形態におけるクエリ、SQL及びCQLの一例を示す説明図である。

[図8B]本実施形態におけるクエリ、SQL及びCQLの一例を示す説明図である。

[図9]データ検索モジュールによって実行される処理の概要を説明するフローチャートである。

[図10A]本実施形態においてクエリ解析部によって切り出される文字列を示す説明図である。

[図10B]本実施形態のクエリ解析部によって実行されるクエリ解析処理の一例を説明するフローチャートである。

[図11A]本実施形態におけるSQLを生成するためのSQLテンプレートを示す説明図である。

[図11B]本実施形態のSQL生成部によって実行される処理の一例を説明するフローチャートである。

[図12A]本実施形態におけるCQLを生成するためのCQLテンプレートを示す説明図である。

[図12B]本実施形態において、CQL生成部によって実行される処理を説明するフローチャートである。

[図13]本実施形態のDB検索部によって実行されるSQL処理の一例を説明

するフローチャートである。

[図14A]本実施形態のデータ解凍／並び替え処理部によって実行される処理の一例を説明するフローチャートである。

[図14B]本実施形態のデータ解凍／並び替え処理部によって実行される処理の一例を説明するフローチャートである。

[図15]本実施形態におけるストリームデータ処理部の構成例を示す説明図である。

[図16]本実施形態のユーザ定義関数の一例を示す説明図である。

[図17]本実施形態のユーザ定義関数に含まれる関数によって抽出された所定の区間のデータの一例を示す説明図である。

[図18A]本実施形態の切り出し処理部によって実行される処理の一例を説明するフローチャートである。

[図18B]本実施形態の切り出し処理の詳細を説明するフローチャートである。

[図19]本実施形態の切り出し処理部によって出力される出力結果の一例を示す説明図である。

[図20]本発明の実施形態のデータ出力部によって実行される処理の一例を説明するフローチャートである。

[図21]本実施形態において、特徴値検索処理を実行する場合のクエリ及びSQLの一例を示す説明図である。

[図22]本実施形態において、特徴値検索用のSQLを生成するためのSQLテンプレートを示す説明図である。

[図23]本実施形態の特徴値検索用SQL生成部によって実行される処理の一例を説明するフローチャートである。

[図24]本実施形態において、データ検索モジュールによって実行される処理を並列に実行する場合を説明するフローチャートである。

[図25]本発明の実施形態の特徴値の種類と生成方法を説明する図である。

[図26]本発明の実施形態の時系列データブロックの時間幅を決定する方法を説明する図である。

[図27]本実施形態のストリームデータ処理部によって実行される処理の一例を説明するフローチャートである。

[図28]本実施形態の入力制御部によって実行される処理の一例を説明するフローチャートである。

[図29]本実施形態の出力制御部によって実行される処理の一例を説明するフローチャートである。

発明を実施するための形態

[0011] 図1は、本発明が適用されたセンサネットワークシステムの一実施形態における計算機システムの概要を説明するブロック図である。

[0012] センサネットワークシステムは、プラント102、データセンタ105及びプラント監視サイト107を備え、ネットワーク103を介して互いに接続される。ネットワーク103としては、例えば、専用線やいわゆるインターネットなどの広域網、又は、LANなどのローカルなネットワークを用いてよい。

[0013] プラント102は、センサ100及びデータ収集装置101を備える。センサ100とデータ収集装置101とは互いに接続される。なお、接続方法は、直接接続してもよいし、LAN (Local Area Network) 又は無線ネットワーク等を介して接続されていてもよい。

[0014] センサ100は、各種データを検出する。なお、センサ100は目的とするデータを検出するために様々な種類のセンサを含む。

[0015] データ収集装置101は、センサ100によって検出されたデータを収集し、ネットワーク103を介して、センサ100から収集されたデータをデータセンタ105に送信する。データ収集装置101は、CPU (図示省略) と、CPUに接続されるメモリ (図示省略)、CPUに接続されるネットワークインタフェース (図示省略) 及び記憶媒体 (図示省略) を備える。なお、データ収集装置101は、ディスプレイ又は入力装置等を備えてもよい。

[0016] データセンタ105は、複数のプラント102のデータ収集装置101か

ら送信されるデータを一元管理するセンタサーバ１０４を備える。

[0017] センタサーバ１０４は、データ収集装置１０１から送信されたデータを格納する。またセンタサーバ１０４は、プラント監視サイト１０７等を介して管理者からの要求に基づいて、格納されたデータから所定のデータを取り出し、又は、取り出されたデータを解析して解析結果を出力する。

[0018] センタサーバ１０４の具体的な装置構成については、図２を用いて後述する。

[0019] プラント監視サイト１０７は、データセンタ１０５に各種問い合わせを行うためのデータ表示端末１０６を備える。

[0020] データ表示端末１０６は、ネットワーク１０３を介して、センタサーバ１０４に格納されたデータを監視又は閲覧する。データ表示端末１０６は、ＣＰＵ（図示省略）と、ＣＰＵに接続されるメモリ（図示省略）、ＣＰＵに接続されるネットワークインタフェース（図示省略）、記憶媒体（図示省略）、表示装置（図示省略）及び入力装置（図示省略）を備える。

[0021] 図２は、本実施形態におけるセンタサーバ１０４のハードウェア構成を説明するブロック図である。

[0022] センタサーバ１０４は、ＣＰＵ１２１、メモリ１２２、ＨＤＤ１２３、ディスプレイ１２４、ネットワークインタフェース１２５、マウス１２６、キーボード１２７、電源装置１２８を備える。

[0023] ＣＰＵ１２１は、メモリ１２２上に展開されたプログラムを実行する。

[0024] メモリ１２２は、ＣＰＵ１２１によって実行されるプログラム及び当該プログラムを実行するために必要となる情報を格納する。また、メモリ１２２は、ＣＰＵが各種処理を実行するためのワークエリアとして用いられる。本実施形態では、メモリ１２２上にＯＳ１５８（図３参照）、データロードモジュール１５３（図３参照）、データベース１５４（図３参照）及びデータ検索モジュール１５７（図３参照）等のプログラムがＨＤＤ１２３よりロードされ実行される。

[0025] ＨＤＤ１２３は、メモリ１２２に読み出されるプログラム及び各種情報を

格納する。図2に示す例では、HDD123は一つのみであるが、複数のHDD123を備えていてもよいし、複数のHDD123から構成されるアレigroupであってもよい。

[0026] ネットワークインタフェース125は、ネットワーク103と接続するためのインタフェースである。

[0027] 電源装置128は、センタサーバ104に電力を供給するための装置である。

[0028] ディスプレイ124は、センタサーバ104を操作する操作者に各種情報を表示する装置である。マウス126は、センタサーバ104を操作する操作者が利用するポインティングデバイスである。キーボード127は、センタサーバ104を操作する操作者が利用する入力装置である。

[0029] 図3は、本実施形態におけるセンタネットワークシステムのソフトウェア構成を説明するブロック図である。

[0030] センタサーバ104上では、OS (Operating System) 158がCPU121によって実行される。OS158は、各種処理を実行する。具体的には、OS158は、データローダモジュール153、データベース154及びデータ検索モジュール157を実行する。

[0031] データベース154は、データ収集装置101から送信され、データローダモジュール153によって所定の形式に変換されたデータを管理する管理ソフトウェアである。データベース154に格納されるデータの詳細については、図4を用いて後述する。

[0032] データローダモジュール153は、プラント102に含まれるデータ収集装置101からネットワーク103を介してデータを受信し、受信したデータを所定の形式に変換してデータベースに格納するソフトウェアである。

[0033] 図3に示す例では、データローダモジュール153は、データ収集装置101からファイル名が「s i d . c s v」の入力ファイル1 (151)、及び、ファイル名が「i n p u t . c s v」の入力ファイル2 (152)を受信し、入力ファイル1 (151) 及び入力ファイル2 (152)を所定の形

式に変換してデータベース 154 に格納する。

- [0034] なお、図 3 に示すファイル 1 (151) 及びファイル 2 (152) は、CSV (Comma Separated Value) 形式のファイルである。ここで、CSV ファイルとは、複数のフィールドが「, (カンマ)」で区切られたファイル形式を表す。図 3 では CSV 形式を使用する例を記したが、カンマの代わりに TAB で区切られたファイル形式 (TSV 形式) やバイナリ形式であってもよい。
- [0035] データ検索モジュール 157 は、入力されたクエリに基づいて、目的とするデータをデータベース 154 から検索し、検索結果を出力するソフトウェアである。
- [0036] 図 3 に示す例では、データ検索モジュール 157 は、データ表示端末 106 を操作するユーザ 150 からクエリ 155 を受け付け、当該クエリ 155 に基づいてデータを検索し、当該データ検索の結果を検索結果ファイル 156 として出力する。データ表示端末 106 は、検索結果ファイル 156 を読み出し、画面にデータを数字、又は、グラフィカルな波形等の形式に変換して表示する。
- [0037] 図 4 は、本実施形態におけるデータローダモジュール 153 及びデータベース 154 を説明するブロック図である。
- [0038] はじめに、入力ファイル 1 (151)、入力ファイル 2 (152) 及びデータベース 154 に格納されるデータの形式について説明する。
- [0039] 入力ファイル 1 (151) は、Name カラム 1511 及び ID カラム 1512 を含む。
- [0040] Name カラム 1511 は、プラント 102 に備わるセンサ 100 のセンサ名である。ID カラム 1512 は、プラント 102 に備わるセンサ 100 を一意に識別するための識別子である。
- [0041] 入力ファイル 2 (152) は、Date time カラム 1521 及びセンサ名カラム 1522 を含む。
- [0042] Date time カラム 1521 は、センサ 100 がデータを取得した時

刻を示す時刻情報である。センサ名カラム 1522 は、各センサ 100 の名称から構成され、Name カラム 1511 と対応するセンサ名が格納される。センサ名カラム 1522 を構成する各センサ 100 には、各センサ 100 が検出した値が格納される。

[0043] 図 4 に示す例では、入力ファイル 2 (152) は、各センサ 100 が 1 秒ごとに検出したデータを格納する。

[0044] データベース 154 は、SID テーブル 207 及び DAT テーブル 208 を格納する。

[0045] SID テーブル 207 は、Name カラム 2071 及び ID カラム 2072 を含む。

[0046] Name カラム 2071 は、プラント 102 に備わるセンサ 100 のセンサ名である。ID カラム 2072 は、プラント 102 に備わるセンサ 100 を一意に識別するための識別子である。

[0047] DAT テーブル 208 は、入力ファイル 2 (152) に含まれるデータを各センサ 100 について所定の連続する時間単位 (時系列データブロック) 毎に圧縮したデータを格納する。具体的には、1 行毎に、各センサ 100 について 1 時間分の圧縮されたデータが格納される。

[0048] DAT テーブル 208 は、Date time カラム 2081、ID カラム 2082、CDATA カラム 2083、Max Val カラム 2084 及び Min Val カラム 2085 を含む。

[0049] Date time カラム 2081 は、圧縮された時系列データブロックの先頭時刻を示す時刻情報を格納する。ここで、先頭時刻とは、Date time カラム 1521 のうち、最も過去の時刻、つまり、圧縮される時系列データブロックの起点となる時刻を表す。

[0050] ID カラム 2082 は、センサ 100 を識別するための識別子である。CDATA カラム 2083 は、圧縮されたデータである。

[0051] Max Val カラム 2084 は、圧縮された時系列データブロックに含まれるデータの最大値である。Min Val カラム 2085 は、圧縮された時

系列データブロックに含まれる最小値である。

- [0052] 図4に示す例では、データ群201について圧縮された時系列データブロックが行209に対応し、データ群202について圧縮された時系列データブロックが行210に対応する。
- [0053] 例えば、行209は、センサ名が「S4」のセンサ100について「2009/10/1の0時0分0秒」から1時間分のデータが圧縮されたデータ「DDDD」であり、圧縮前のデータに含まれる最大値は「99」、圧縮前のデータに含まれる最小値は「52」であることを示す。
- [0054] 本実施形態では、入力ファイル2（152）の3600行（1時間）のデータが1行に圧縮される。なお、本実施形態の以降の説明では、時系列データブロックに格納される時間範囲は1時間固定とするが、時系列データブロックの時間範囲を1時間以外に変更することも可能である。
- [0055] 時系列ブロックの時間範囲の決定方法としては、利用者の最も利用する検索期間を用いる。例えば、最も多い期間検索の範囲の1/10の期間を時系列データブロックの時間範囲とする。これによって、検索範囲が1日であれば時系列データブロックの時間範囲は2.4時間、検索範囲が1ヶ月であれば時系列データブロックの時間範囲は3日とする。これは、データ検索モジュール157のクエリ解析部において、図8Aのクエリ例1のwhere__time_range句で指定された検索期間から時間範囲を算出し、時間範囲を頻度表として管理することによって実現することが可能となる。
- [0056] また、時系列ブロックの時間範囲の決定方法としては、時系列の特徴変化が起こる最小期間を用いることもできる。図26を用いて、時系列データブロック2601を分割していき、時系列ブロック2605を最適な時系列ブロックとする方法について説明する。
- [0057] 図26は、本発明の実施形態の時系列データブロックの時間幅を決定する方法を説明する図である。
- [0058] 時系列ブロック2601において、特徴値a[1][1]を得る。次に時系列データブロックを半分に分割し、特徴値a[2][1]、a[2][2]

」を得る。ここで特徴値 $a[i][j]$ において、 i は何回分割したかを示す分割レベル、 j は何番目の時系列データブロックの特徴値かを示す番号を表す。前述した処理を再帰的に繰り返す。

[0059] ここで、分割レベル i における特徴値と分割レベル $i+1$ における特徴値との差が閾値以下であるところを時系列データブロックの最適な分割レベルとし、その時系列データブロックの幅を最適な時間範囲とする。特徴値 $a[i][j]$ と $a[i+1][j]$ 、 $a[i][j]$ と $a[i+1][j+1]$ の差の大きい方の値を、すべての時系列データブロック j に対し平均を取り、この値を「特徴値との差」として閾値判定することも可能である。

[0060] 図4に示すように、圧縮データを格納することによって、データベースの使用容量を削減することが可能となる。

[0061] 次に、データローダモジュール153について説明する。

[0062] データローダモジュール153は、データ集約部203、特徴値抽出部204、特徴値抽出用ストリームデータ処理部2041、データ圧縮部205及びデータ挿入部206を備える。

[0063] データ集約部203は、各センサ100毎に所定の時間間隔（本実施形態では1時間）のデータを集約する。

[0064] 特徴値抽出部204は、圧縮されているデータの中から特徴値を抽出する。特徴値は、データベース154から圧縮データを検索するときに用いられる。特徴値を用いた検索方法については、図23を用いて後述する。なお、本実施形態では、圧縮されるデータの中から最大値及び最小値が特徴値として抽出される。なお、抽出される特徴値は、平均又は分散等圧縮されるデータの特徴を示す値であれば、どのようなものであってもよい。

[0065] 特徴値の例について図25を用いて説明する。

[0066] $i=1$ のケースでは、圧縮されているデータの始点、終点等の特定の値を特徴値として使用する。

[0067] $i=2$ のケースでは、圧縮されているデータの全体ないし任意の区間の時系列に対する最大値、最小値、平均値、分散値、又は標準偏差値を特徴量とし

て使用する。

- [0068] i_3 及び i_4 のケースでは、複数のセンサ観測値 i_3 、 i_4 の平均値を特徴値として使用する。
- [0069] i_5 のケースでは、圧縮されているデータの全体ないし任意の区間をフレーム化し、FFT (Fast Fourier Transform) 変換を実施することによって周波数帯域に変換し、あらかじめ指定された周波数 A、B、C での振幅を特徴値として使用する。
- [0070] 特徴値抽出部 204 は、特徴値抽出用ストリームデータ処理部 2041 を利用することによって特徴値を生成することができる。すなわち、リアルタイムに到来する時系列データを所定期間メモリ上に蓄積し、最大値、最小値、平均値、又は分散値等の時系列解析を行うことによって特徴値を生成することができる。
- [0071] データ圧縮部 205 は、各センサ毎に所定の時間間隔のデータを圧縮する。データ挿入部 206 は、圧縮データを DAT テーブル 208 に格納する。
- [0072] 図 5A は、本実施形態において、データローダモジュール 153 によって実行される処理を説明するフローチャートである。
- [0073] データローダモジュール 153 は、データベース 154 に SID テーブル 207 が存在するか否かを判定する (S250)。例えば、データローダモジュール 153 が、データベース 154 に SID テーブル 207 の存在を問い合わせる SQL を発行する等によって判定できる。
- [0074] データベース 154 に SID テーブル 207 が存在すると判定された場合、データローダモジュール 153 は、S252 に進む。
- [0075] データベース 154 に SID テーブル 207 が存在しないと判定された場合、データローダモジュール 153 は、入力ファイル 1 (151) を読み出し、SID テーブル 207 を生成する (S251)。
- [0076] データローダモジュール 153 は、入力ファイル 2 (152) から 1 行分の文字列を読み出し、カンマを区切り記号として読み出された文字列からカラムに相当する文字列を抽出し、配列 csv に格納する (S252)。例え

ば、 c_i 個の文字列が抽出された場合、抽出された文字列は、それぞれ、配列 $csv[0] \sim csv[c_i - 1]$ に格納される。ここで、 c_i はカラムの数を表す。本実施形態では、 $csv[0]$ には `datetime` が格納され、 $csv[1] \sim csv[c_i - 1]$ にはセンサの値が格納される。

[0077] データローダモジュール 153 は、入力ファイル 2 (152) のセンサ名カラム 1522 から ID カラム 2082 が検索できるようにするための配列 `id` を生成する (S253)。

[0078] データローダモジュール 153 は、圧縮データを作成するためのデータ配列を定義する (S254)。本実施形態では、1つのセンサ 100 の 1 時間のデータが圧縮されるため、 $d[c_i][3600]$ 配列が定義される。すなわち、1つのセンサ 100 について 3600 個の配列が定義される。

[0079] データローダモジュール 153 は、データベース 154 に DAT テーブル 208 が存在するか否かを判定する (S255)。例えば、データローダモジュール 153 が、データベース 154 に DAT テーブル 208 の存在を問い合わせる SQL を発行する等によって判定できる。

[0080] データベース 154 に DAT テーブル 208 が存在すると判定された場合、データローダモジュール 153 は、S257 に進む。

[0081] データベース 154 に DAT テーブル 208 が存在しないと判定された場合、データローダモジュール 153 は、DAT テーブル 208 を生成する (S256)。

[0082] データローダモジュール 153 は、S254 において取得された配列から圧縮データを生成して、DAT テーブル 208 に格納するためのデータロード処理を実行する (S257)。データロード処理の詳細については、図 5B を用いて後述する。

[0083] データローダモジュール 153 は、入力ファイル 2 (152) の全てのデータについて処理を終了したか否かを判定する (S258)。

[0084] 入力ファイル 2 (152) の全てのデータについて処理を終了していないと判定された場合、データローダモジュール 153 は、S257 に戻り同様

の処理を実行する。

- [0085] 入力ファイル 2 (152) の全てのデータについて処理を終了したと判定された場合、データローダモジュール 153 は、処理を終了する。
- [0086] 図 5 B 及び図 5 C は、本実施形態において、データローダモジュール 153 によって実行されるデータロード処理を説明するフローチャートである。
- [0087] データローダモジュール 153 は、時間 (秒) を示す変数 i を初期化し、全ての時間 (秒) について処理を実行する (S259)。具体的には、データローダモジュール 153 は、変数 i を「0」に設定し、変数 i が「3600」より大きくなるまで繰り返し処理を実行する。なお、本実施形態では、0～3599 秒までのデータが処理対象となる。
- [0088] データローダモジュール 153 は、入力ファイル 2 (152) から 1 行分のデータを読み出し、読み出されたデータから c_i 個の文字列を抽出し、配列 $c_{sv}[0] \sim c_{sv}[c_i - 1]$ に格納する (S260)。
- [0089] データローダモジュール 153 は、変数 i が「0」であるか否かを判定する (S261)。
- [0090] 変数 i が「0」でないと判定された場合、データローダモジュール 153 は、S263 に進む。
- [0091] 変数 i が「0」であると判定された場合、データローダモジュール 153 は、DAT テーブル 208 の $Date\ time$ カラム 2081 に $c_{sv}[0]$ を格納する (S262)。
- [0092] データローダモジュール 153 は、各センサ 100 毎にデータを集約する (S263)。具体的には、データ集約部 203 は、配列 c_{sv} に格納される値を数値化して配列 d に設定する。
- [0093] データローダモジュール 153 は、繰り返し処理を実行するか、すなわち、全ての時間について処理が終了したか否かを判定する (S264)。
- [0094] 全ての変数 i について処理が終了していないと判定された場合、データローダモジュール 153 は、S259 に戻り、変数 i を「1」加算して、S260～S263 までの処理を実行する。

- [0095] 全ての変数 i について処理が終了したと判定された場合、データローダモジュール 153 は、センサ 100 の数を表す変数 j を初期化し、全てのセンサ 100 について処理を実行する (S265)。具体的には、データローダモジュール 153 は、変数 j に「1」を設定し、変数 j が c_n より大きくなるまで繰り返し処理を実行する。
- [0096] データローダモジュール 153 は、配列 $d[j][i]$ の中から特徴値を抽出する (S266)。具体的には、特徴値抽出部 204 が、配列 $d[j][0] \sim d[j][3599]$ の中から最大値、最小値、平均値、分散値若しくは周波数スペクトル上の特定値のいずれか、又はこれらの組み合わせを抽出する。
- [0097] データローダモジュール 153 は、配列 $d[j][i]$ を所定の形式に圧縮する (S267)。具体的には、データ圧縮部 205 が、配列 $d[j][0] \sim d[j][3599]$ を所定の形式に圧縮する。なお、圧縮形式としては、zip、lzh、gzip 及び bzip2 等が考えられる。
- [0098] データローダモジュール 153 は、圧縮されたデータを DAT テーブル 208 に格納する (S268)。具体的には、データ挿入部 206 が、S253 において生成された $id[i]$ を ID カラム 2082 に、S262 において設定された $csv[0]$ を Datetime カラム 2081 に、S266 において抽出された最大値を DAT テーブル 208 の MaxVal カラム 2084 に、S266 において抽出された最小値を MinVal カラム 2085 に、さらに、S267 において圧縮されたデータを CDATA 2083 に格納する。
- [0099] データローダモジュール 153 は、全てのセンサ 100 について処理が終了したか否かを判定する (S269)。
- [0100] 全てのセンサ 100 について処理が終了していないと判定された場合、データローダモジュール 153 は、S266 に戻り同様の処理を実行する。
- [0101] 全てのセンサ 100 について処理が終了したと判定された場合、データローダモジュール 153 は、処理を終了する。

- [0102] 図6は、本実施形態におけるデータ検索モジュール157を説明するブロック図である。
- [0103] データ検索モジュール157は、入力されたクエリ155に基づいて、データベース154にSQLを発行し、発行されたSQLに対する応答に基づいて検索結果を生成し、生成された検索結果を検索結果ファイル156に出力する。
- [0104] データ検索モジュール157は、クエリ解析部300、SQL生成部301、特徴値検索用SQL生成部302、CQL生成部303、DB検索部304、データ解凍／並び替え処理部305、ストリームデータ処理部306、切り出し処理部307及びデータ出力部308を備える。
- [0105] クエリ解析部300は、入力されたクエリ155を解析し、当該クエリ155の内容をグローバル変数309として出力する。なお、グローバル変数309は、メモリ122等の記憶領域に格納され、SQL生成部301、特徴値検索用SQL生成部302、CQL生成部303、DB検索部304、データ解凍／並び替え処理部305、ストリームデータ処理部306、切り出し処理部307及びデータ出力部308の各部が参照することができる。
- [0106] SQL生成部301は、クエリ解析部300の解析結果であるグローバル変数309に基づいて、データベース154を検索するためのSQL310を生成し、生成されたSQL310をDB検索部304に出力する。
- [0107] 特徴値検索用SQL生成部302は、クエリ解析部300の解析結果であるグローバル変数309に基づいて、特徴値を用いてデータベース154を検索するためのSQL310を生成し、生成されたSQL310をDB検索部304に出力する。
- [0108] CQL生成部303は、クエリ解析部300の解析結果であるグローバル変数309に基づいて、ストリームデータ処理を実行するためのCQL311を生成し、生成されたCQL311をストリームデータ処理部306に出力する。
- [0109] DB検索部304は、SQL生成部301又は特徴値検索用SQL生成部

302によって生成されたSQL310をデータベース154へ発行する。DB検索部304は、SQL310の実行結果、すなわち、データ検索結果をファイルA312に出力する。なお、ファイルA312に出力されるデータは圧縮データである。

[0110] データ解凍／並び替え処理部305は、ファイルA312を読み出し、圧縮データを解凍し、かつ、解凍されたデータを所定の条件に基づいて並び替える。データ解凍／並び替え処理部305は、処理されたデータをファイルB313に出力する。

[0111] ストリームデータ処理部306は、CQL生成部303によって生成されたCQL311に基づいて、ファイルB313を入力データとして読み出し、読み出された各データに対してストリームデータ処理を実行する。ストリームデータ処理部306は、ファイルC314に処理されたデータを出力する。

[0112] 切り出し処理部307は、ファイルC314から所定の条件に一致するデータを抽出する。切り出し処理部307は、ファイルD315に抽出されたデータを出力する。

[0113] データ出力部308は、ファイルD315に格納されるデータを用いて出力結果を生成し、生成された出力結果を検索結果ファイル156として出力する。

[0114] 図7は、本実施形態において、データ検索モジュール157によって出力されるデータの構造の一例を示す説明図である。

[0115] データ構造1(340)は、ファイルA312に格納されるデータ(パケット)のデータ構造である。ファイルA312は複数のパケットを格納する。

[0116] データ構造1(340)は、`datetime3401`、`idx3402`、`zlen3403`及び`zblk3404`から構成される。

[0117] `datetime3401`は、日付及び時間等の時刻を表し、DATテーブル208の`Datetime`カラム2081に対応する。本実施形態では

、`datetime3401`のデータ長は`8 byte`である。

[0118] `idx3402`は、センサ100を識別するためのセンサ識別番号を表す。クエリ例1の`input__item`句に記されるセンサの並び順がセンサ識別番号として割り当てられる。本実施形態では、`idx3402`のデータ長は`4 byte`である。

[0119] `zlen3403`は、圧縮されたデータのブロック長を表し、本実施形態ではデータ長は`4 byte`である。

[0120] `zblk3404`は、圧縮されたデータを表す。具体的には、圧縮されたデータそのものである。本実施形態では、`zblk3404`のデータ長は`n byte`である。

[0121] データ構造2 (341) は、ファイルB313、ファイルC314及びファイルD315に格納されるデータ (バイナリ) のデータ構造である。

[0122] `datetime3411`は、日付及び時間等の時刻を表す。本実施形態では、`datetime3411`のデータ長は`8 byte`である。

[0123] `valn3412`は、データ3413の数を表し、本実施形態では、データ長は`8 byte`である。

[0124] データ3413は、データ解凍／並び替え処理部305、ストリームデータ処理部306及び切り出し処理部307によって算出された値を表し、本実施形態では、データ長はそれぞれ、`4 byte`である。

[0125] 例えば、「2009/10/1 00:00:00, 4, 100, 300, 12, 52」というデータがあった場合、`datetime3411`には「2009/10/1 00:00:00」が、`valn3412`には「4」が、`valn[0]`には「100」が、`valn[1]`には「300」が、`valn[2]`には「12」が、`valn[3]`には「52」が格納される。

[0126] データ構造3 (342) は、検索結果ファイル156に格納されるデータのデータ構造である。

[0127] データ構造3 (342) には、日付及び時間等の時刻、並びに、各センサ

100の値が格納される。

[0128] 図8A及び図8Bは、本実施形態におけるクエリ155、SQL310及びCQL311の一例を示す説明図である。

[0129] クエリ例1(350)及びクエリ例2(351)は、クエリ155の一例である。

[0130] クエリ例1(350)は、ストリームデータ処理が実行される場合のクエリ155である。クエリ例1(350)は、センサの名称が「Sensor1」及び「Sensor2」のセンサ100のデータに対して、where__timerange句及びwhere__condition句において指定された条件を満たすデータを抽出し、さらに、抽出されたデータを「result.csv」に格納することを示す。

[0131] クエリ例2(351)は、ストリームデータ処理が実行されない場合のクエリ155である。クエリ例2(351)は、センサの名称が「Sensor1」、「Sensor2」及び「Sensor3」のセンサ100のデータに対して、where__timerange句において指定された時間間隔のデータを抽出し、抽出されたデータを「result.csv」に格納することを示す。

[0132] SQL例1(352)は、クエリ例1(350)が入力された場合に、SQL生成部301によって生成されるSQL310の一例である。なお、SQL310の生成方法については、図11A及び図11Bを用いて後述する。

[0133] CQL例1(353)は、クエリ例1(350)が入力された場合に、CQL生成部303によって生成されるCQL311の一例である。なお、CQL311の生成方法については、図12A及び図12Bを用いて後述する。

[0134] 図9は、本実施形態のデータ検索モジュール157によって実行される処理の概要を説明するフローチャートである。

[0135] データ検索モジュール157は、データ表示端末106からクエリ155

の入力を受け付け、処理を開始する。

- [0136] データ検索モジュール157は、入力されたクエリ155を解析する（S320）。具体的には、クエリ解析部300が入力されたクエリ155を解析する。
- [0137] データ検索モジュール157は、入力されたクエリ155が特徴値検索の実行を要求するクエリであるか否かを判定する（S321）。具体的には、クエリ解析部300が、入力されたクエリ155に`meta__search`句が含まれるか否かを判定する。入力されたクエリ155に`meta__search`句が含まれる場合、入力されたクエリ155が特徴値検索の実行を要求するクエリであると判定される。なお、`meta__search`句を含むクエリ155の例については、図22を用いて後述する。
- [0138] 入力されたクエリ155が特徴値検索の実行を要求するクエリでないと判定された場合、データ検索モジュール157は、SQL310を生成する（S322）。具体的には、SQL生成部301がSQL310を生成し、S324に進む。
- [0139] 入力されたクエリ155が特徴値検索の実行を要求するクエリであると判定された場合、データ検索モジュール157は、特徴値検索用のSQL310を生成する（S323）。具体的には、特徴値検索用SQL生成部302が特徴値検索用のSQL310を生成し、S324に進む。
- [0140] 次に、データ検索モジュール157は、ストリームデータ処理を実行するか否かを判定する（S324）。具体的には、クエリ解析部300が、入力されたクエリ155に`select__items`句が含まれるか否かを判定する。入力されたクエリ155に`select__items`句が含まれる場合には、ストリームデータ処理を実行すると判定される。
- [0141] ストリームデータ処理を実行しないと判定された場合、データ検索モジュール157は、SQL生成部301又は特徴値検索用SQL生成部302によって生成されたSQL310に基づいて、データベース154を検索する（S325）。具体的にはDB検索部304が、SQL生成部301又は特

徴値検索用SQL生成部302によって生成されたSQL310に基づいて、データベース154を検索し、検索された圧縮データをファイルA312に出力する。

[0142] データ検索モジュール157は、ファイルA312に格納される圧縮データを解凍し、解凍されたデータを並び替える(S326)。具体的には、データ解凍／並び替え処理部305が、ファイルAに格納される圧縮データを解凍し、解凍されたデータを並び替えてファイルBに出力し、S332に進む。

[0143] S324において、ストリームデータ処理を実行すると判定された場合、データ検索モジュール157は、CQL311を生成する(S327)。具体的には、CQL生成部303がCQL311を生成する。

[0144] データ検索モジュール157は、SQL生成部301又は特徴値検索用SQL生成部302によって生成されたSQL310に基づいて、データベース154を検索する(S328)。

[0145] データ検索モジュール157は、ファイルA312に格納された圧縮データを解凍し、解凍されたデータを並び替える(S329)。当該処理は、S326と同一の処理である。

[0146] データ検索モジュール157は、CQL生成部303によって生成されたCQL311に基づいて、ストリームデータ処理を実行する(S330)。具体的には、ストリームデータ処理部306が、CQL生成部303によって生成されたCQL311に基づいて、ストリームデータ処理を実行し、実行結果をファイルC314に出力する。

[0147] データ検索モジュール157は、ファイルC314に格納されるデータから出力対象となるデータを切り出す(S331)。具体的には、切り出し処理部307が、ファイルC314に格納されるデータから出力対象となるデータを切り出し、切り出されたデータをファイルD315に出力して、S332に進む。

[0148] データ検索モジュール157は、ファイルB又はファイルD315に格納

されるデータをデータ表示端末 106 に表示するための出力データに変換して、検索結果ファイル 156 に出力し（S332）、処理を終了する。

[0149] 以下、各ステップの処理の詳細を説明する。

[0150] 図 10A は、本実施形態においてクエリ解析部 300 によって切り出される文字列を示す説明図である。

[0151] 文字切り出し部分 360 は、入力されたクエリ 155 から切り出される文字列を表す。クエリ解析部 300 が、切り出しキーワードを指示するテンプレートである文字切り出し部分 360 を保持する。

[0152] 図 10A に示す例では、9 つのキーワードから変数「\$input」、「\$range」、「\$select」、「\$start」、「\$end」、「\$cond」、「\$prev」、「\$post」、「\$step」、「\$file」及び「\$meta」が切り出される。ここで、キーワードとは、「input__items:」及び「select__items」等を表す。

[0153] 例えば、入力されたクエリ 155 からキーワード「input__items:」以下の文字列が切り出され、変数「\$input」に切り出された文字列が格納される。

[0154] 以下、クエリ解析部 300 が実行するクエリ解析処理について説明する。

[0155] 図 10B は、本実施形態のクエリ解析部 300 によって実行されるクエリ解析処理の一例を説明するフローチャートである。

[0156] クエリ解析部 300 は、クエリ 155 の入力を受け付け、処理を開始する。

[0157] まず、クエリ解析部 300 は、変数を初期化する（S361）。具体的には、クエリ解析部 300 は、「\$range」、「\$cond」及び「\$step」に「1」を設定し、「\$prev」及び「\$post」に「0」を設定する。また、クエリ解析部 300 は、「\$input」、「\$select」、「\$start」、「\$end」、「\$file」及び「\$meta」についてはNull初期化を実行する。

- [0158] 次に、クエリ解析部300は、文字切り出し部分360に基づいて、入力されたクエリ155から文字列を切り出し、切り出された文字列を各変数に格納する(S362)。
- [0159] 具体的には、クエリ解析部300は、クエリ155に含まれる9つの句から文字列を切り出し、切り出された文字列を、それぞれ、変数「\$input」、「\$range」、「\$select」、「\$start」、「\$end」、「\$cond」、「\$prev」、「\$post」、「\$step」、「\$file」及び「\$meta」に格納する。
- [0160] 例えば、クエリ例1(350)の場合、変数「\$input」には「'Sensor1'」、「'Sensor2'」が格納される。
- [0161] なお、クエリ155に含まれないキーワードについては、S361において設定された値がそのまま用いられる。
- [0162] クエリ解析部300は、変数「\$input」及び「\$select」に含まれる要素の数を算出する(S363)。
- [0163] 具体的には、クエリ解析部300は、変数「\$input」及び「\$select」に含まれる文字列において、「, (カンマ)」で区切られた要素の数をそれぞれカウントすることによって、要素の数を算出する。さらに、クエリ解析部300は、変数「\$input」に含まれる要素数を変数「\$inum」に代入し、変数「\$select」に含まれる要素数を変数「\$snum」に格納する。
- [0164] 例えば、変数「\$input」に「'Sensor1'」、「'Sensor2'」が格納されている場合、「\$inum」には「2」が格納される。
- [0165] クエリ解析部300は、各変数をグローバル変数309として出力し(S364)、処理を終了する。
- [0166] 図11Aは、本実施形態におけるSQL310を生成するためのSQLテンプレートを示す説明図である。
- [0167] SQLテンプレート370は、SQL310を生成するためのテンプレートであり、SQL生成部301が保持する。

[0168] 図11Aに示す例では、SIDテーブル207とDATテーブル208との両テーブルをそれぞれのIDカラム(2072、2082)に示される値で等価結合(joinオペレーション)したテーブルから、時刻(Date time)、センサ名(SID. Name)及び圧縮データ(DAT. CDATA)を所定の条件に基づいて検索するSQLであることを示す。また、所定の条件は、where句以下に定義されている。すなわち、データを時間順に並び替え、指定されたセンサ名及び指定された時間区間に一致するデータを取得するSQLが定義される。

[0169] SQL生成部301は、グローバル変数309を用いて、SQLテンプレート370の太文字及び下線部で示された部分に必要なデータを代入することによって、SQL310を生成する。具体的には、SQL生成部301は、「\$db__input」、「\$db__start」及び「\$db__end」を生成して、生成されたデータをSQLテンプレート370に代入する。

[0170] 以下、SQL生成処理について説明する。

[0171] 図11Bは、本実施形態のSQL生成部301によって実行される処理の一例を説明するフローチャートである。

[0172] SQL生成部301は、グローバル変数309を用いて、「\$db__input」、「\$db__start」及び「\$db__end」を生成する(S371)。

[0173] 具体的には、「\$db__input」及び「\$db__end」については、「\$db__input」=「\$input」及び「\$db__end=\$end」と生成される。また、「\$db__start」については、「\$db__start=\$start-\$range」を計算し、さらに、当該計算結果の分及び秒単位の値が「0」に初期化される。これは、本実施形態では1時間単位に圧縮されたデータを検索するためである。

[0174] 次に、SQL生成部301は、SQL310を生成し(S372)、処理を終了する。

[0175] 具体的には、SQL生成部301は、生成された「\$db__input」

、「\$db__start」及び「\$db__end」をSQLテンプレート370に代入することによってSQL310を生成する。

[0176] 図12Aは、本実施形態におけるCQL311を生成するためのCQLテンプレートを示す説明図である。

[0177] CQLテンプレート380は、CQL311を生成するためテンプレートであり、CQL生成部303が保持する。

[0178] CQL生成部303は、グローバル変数309を用いて、CQLテンプレート380の太文字及び下線部で示された部分に必要なデータを代入することによって、CQL311を生成する。

[0179] 具体的には、CQL生成部303は、「\$cql__input」、「\$cql__select」、「\$range」、「\$cql__label」及び「\$cond」を生成して、生成されたデータをCQLテンプレート380に代入する。

[0180] 以下、CQL生成処理について説明する。

[0181] 図12Bは、本実施形態において、CQL生成部303によって実行される処理を説明するフローチャートである。

[0182] CQL生成部303は、グローバル変数309を用いて、「\$cql__input」を生成する(S381)。具体的には、以下の二つの処理が実行される。

[0183] まず、CQL生成部303は、グローバル変数309に含まれる「\$input」を用いて、\$input分解処理385を実行する。すなわち、CQL生成部303は、「\$input」を複数の「\$input__item」に分解する。これは、複数のセンサ名が含まれる「\$input」から、個々のセンサ名を取得するための処理である。

[0184] 次に、CQL生成部303は、「\$input__item」を用いて、\$cql__input生成処理386を実行する。CQL生成部303は、「\$input__item」を用いて、\$cql__input生成処理386の右辺に示すようにして「\$cql__input」を生成する。

- [0185] CQL生成部303は、グローバル変数309を用いて、「\$cq1__select」を生成する(S382)。具体的には、以下の二つの処理が行われる。
- [0186] まず、CQL生成部303は、グローバル変数309に含まれる「\$select」を用いて、\$select分解処理387を実行する。これは、\$selectに含まれる要素を分解するための処理である。CQL生成部303は、\$select分解処理387を実行することによって、「\$func1」又は「\$func2」及び「\$label」を取得する。
- [0187] 次に、CQL生成部303は、「\$func1」又は「\$func2」及び「\$label」を用いて、\$cq1__select生成処理388を実行し、「\$cq1__select」を生成する。ここで「\$func1」は集約関数であり、移動平均(average)、分散、最大値、又は最小値などが記載できる。
- [0188] ここで、CQLでは、average等の集約関数と、「Sensor1」等の信号名とを混在させて扱うことができない。そのため、「\$select」の含まれる要素をlast()関数を用いて集約関数に変換する必要がある。例えば、文字列「'Sensor1' as LABEL」は「last('Sensor1') as LABEL」と集約関数に変換される。
- [0189] CQL生成部303は、グローバル変数309を用いて、「\$cq1__label」を生成する(S383)。具体的には、CQL生成部303は、「\$label」を用いて、\$cq1__label生成処理389を実行することによって、「\$cq1__label」を生成する。
- [0190] CQL生成部303は、S381~S383において生成された「\$cq1__input」、「\$cq1__select」及び「\$cq1__label」、並びに、グローバル変数309に含まれる「\$range」及び「\$cond」をCQLテンプレート380に代入することによって、CQL311を生成する(S384)。
- [0191] 以上のように、データ検索モジュール157は、入力されたクエリ155

に基づいて、SQL 310及びCQL 311を同時に生成することができる。

[0192] 以下、生成されたSQL 310及びCQL 311を用いた処理について説明する。

[0193] 図13は、本実施形態のDB検索部によって実行されるSQL処理の一例を説明するフローチャートである。

[0194] DB検索部304は、グローバル変数309に含まれる「\$input」を用いて、配列KEY[0]～KEY[\$inum-1]を生成する(S400)。

[0195] 具体的には、DB検索部304は、「\$input」を用いて、\$signal抽出処理407を実行する。これによって、「\$inum」個の「\$signal」が抽出される。DB検索部304は、抽出された各「\$signal」を配列KEY[0]～KEY[\$inum-1]に格納する。以上の処理によって、配列KEY[0]～KEY[\$inum-1]が生成される。

[0196] DB検索部304は、生成された配列KEY[0]～KEY[\$inum-1]を用いて、ハッシュ配列hashを生成する(S401)。

[0197] DB検索部304は、データベース154に対してSQL 310を発行し、データベース154がSQL 310を実行する(S402)。これによって、発行されたSQL 310において指定された条件に一致するデータを実行結果として取得することができる。

[0198] DB検索部304は、取得された実行結果から1行分のデータを取り出し、取り出されたデータから日付及び時間等の時刻(Datetime)、センサ名(Name)並びに圧縮データ(CDATA)を取得する(S403)。なお、取り出される1行分のデータは、1つのセンサの1時間分の圧縮されたデータである。

[0199] DB検索部304は、実行結果から取り出された1行分のデータが空である否かを判定する(S404)。すなわち、全ての実行結果について処理が

終了したか否かが判定される。

- [0200] 実行結果から取り出された1行分のデータが空であると判定された場合、DB検索部304は、処理を終了する。
- [0201] 実行結果から取り出された1行分のデータが空でないと判定された場合、DB検索部304は、データ構造1(340)に示すパケットを生成する(S405)。
- [0202] 具体的には、DB検索部304は、取り出された1行分のデータに含まれる日付及び時間等の時刻(Datetime)をdatetime3401に、センサ名(Name)をキーにハッシュ配列hashから得られた値をidx3402に、圧縮データ(CDATA)をzblk3404に、圧縮データ(CDATA)のサイズをzlen3403に格納する。
- [0203] DB検索部304は、生成されたパケットをファイルA312に出力し(S406)、S403に戻り、S403~S406の処理を実行する。
- [0204] 例えば、SQL例1(352)が実行された場合、センサ名が「Sensor1」及び「Sensor2」のセンサ100について「2009-02-01 23:00:00」から「2009-02-20 23:59:59」までの間のデータが複数個のデータ構造1(340)のパケットに変換され、ファイルA312に出力される。
- [0205] 図14A及び図14Bは、本実施形態のデータ解凍／並び替え処理部305によって実行される処理の一例を説明するフローチャートである。
- [0206] データ解凍／並び替え処理部305は、data[3600][sinum]配列、バッファ配列及びblk配列を定義する(S450)。本実施形態では、一つの圧縮データには、1つのセンサ100について1時間分のデータ、3600点が格納されているため、data[3600][sinum]配列が定義される。
- [0207] データ解凍／並び替え処理部305は、グローバル変数309に含まれる「\$snum」が「0」であるか否かを判定する(S451)。これによって、ストリームデータ処理が実行されるか否かが判定される。すなわち、図

8 Aのクエリ例2 (351)に示すようにクエリ155にselect__items句が含まれない場合、「\$snum」は「0」であるため、ストリームデータ処理は実行されないと判定される。

[0208] 「\$snum」が「0」でないと判定された場合、データ解凍／並び替え処理部305は、処理結果を出力するファイルをファイルB313に決定し(S452)、S434に進む。

[0209] 「\$snum」が「0」であると判定された場合、データ解凍／並び替え処理部305は、処理結果を出力するファイルをファイルD315に決定し(S453)、S434に進む。

[0210] データ解凍／並び替え処理部305は、data[3600][\$inum]配列及び時刻を初期化する(S454)。具体的には、データ解凍／並び替え処理部305は、data[3600][\$inum]をNaN(Not a Number)値を用いて初期化し、また、時刻を表す変数lastetを「0」に初期化する。本実施形態では、配列dataの初期化の際、NaN値を用いたが、数値と重複しない値であれば、何でもよい。

[0211] データ解凍／並び替え処理部305は、データ構造1(340)に示すパケットを一つ取り出す(S455)。具体的には、データ解凍／並び替え処理部305は、バッファからパケットを取り出す。バッファにパケットがない場合には、ファイルA312からパケットを取り出す。初回の処理時には、バッファにはパケットが格納されていないため、ファイルA312からパケットが取り出される。

[0212] データ解凍／並び替え処理部305は、取り出されたパケットが空であるか否かを判定する(S456)。すなわち、取り出されたパケットに処理対象のデータが含まれているか否かが判定される。

[0213] 取り出されたパケットが空でないと判定された場合、データ解凍／並び替え処理部305は、当該パケットからデータを抽出する(S457)。具体的には、データ解凍／並び替え処理部305は、パケットから、dattetime3401、idx3402、zlen3403及びzblk3404

を抽出する。

- [0214] データ解凍／並び替え処理部 305 は、同じ時刻の時系列データブロックを一括して処理するため、取り出されたパケットが処理時間範囲内のデータであるか否かを判定する (S458)。具体的には、`latest` には、前回受信したパケットの `datetime3401` が保持されているので、今回受信したパケット (340) の `datetime3401` と `latest` とを比較することによって、当該パケットが処理時間の範囲である 1 時間以内のパケットであるか否かを判定できる。
- [0215] 取り出されたパケットが処理時間範囲内のデータであると判定された場合、データ解凍／並び替え処理部 305 は、圧縮データを解凍する (S459)。具体的には、データ解凍／並び替え処理部 305 は、取り出されたパケットに含まれる `zblk3404` を解凍し、当該結果を `blk` に格納する。
- [0216] データ解凍／並び替え処理部 305 は、解凍されたデータを配列 `data` に格納し (S460)、S455 に戻り S455～S460 の処理を実行する。具体的には、データ解凍／並び替え処理部 305 は、`blk` を配列 `data` に格納する。また、データ解凍／並び替え処理部 305 は、`latest` に `datetime3401` を格納する。
- [0217] S456 において取り出されたパケットが空であると判定された場合、又は、S458 において取り出されたパケットが処理時間範囲内のデータでないと判定された場合、データ解凍／並び替え処理部 305 は、取り出されたパケットをバッファに書き戻す (S461)。
- [0218] データ解凍／並び替え処理部 305 は、ループ回数を示す変数 `i` に「0」に設定し、パケット内の 1 時間分の各データに対して処理 463、処理 464 を繰り返し実行する (S462)。すなわち、変数 `i` が「3600」より大きくなるまで処理が繰り返し実行される。
- [0219] データ解凍／並び替え処理部 305 は、出力結果を生成する (S463)。すなわち、データ解凍／並び替え処理部 305 は、データ構造 2 (341) に示すようなデータを生成する。

- [0220] 具体的には、変数 i に対して、 $latest + i$ を $datetime3411$ に、「 $inum$ 」を $valn3412$ に、配列 $data$ をそれぞれデータ 3413 に格納する。これによって、 $input_items$ 句に指定されるセンサの 1 秒（1 行）のデータに対応する出力結果が得られる。したがって、全ての変数 i について同様の処理を繰り返し実行することによって、 $input_items$ 句に指定されたセンサの 1 時間分のデータが出力結果として得られる。
- [0221] データ解凍／並び替え処理部 305 は、生成された出力結果をファイル B 313 又はファイル D 315 に出力する（S464）。
- [0222] データ解凍／並び替え処理部 305 は、1 時間分のパケットの処理を終了したか否かを判定する（S465）。すなわち、変数 i が「3600」より大きいかが判定される。変数 i が「3600」未満の場合、1 時間分のパケットの処理が終了していないと判定される。
- [0223] 全ての変数 i に対して処理が終了していないと判定された場合、すなわち、変数 i が「3600」未満の場合、データ解凍／並び替え処理部 305 は、S462 に戻り S462～S466 の処理を実行する。
- [0224] 全ての変数 i に対して処理が終了したと判定された場合、すなわち、変数 i が「3600」より大きい場合、データ解凍／並び替え処理部 305 は、ファイル A 312 に格納される全てのパケットについて処理が終了したか否かを判定する（S466）。
- [0225] ファイル A 312 に格納される全てのパケットについて処理が終了していないと判定された場合、データ解凍／並び替え処理部 305 は、S454 に戻り S454～S466 の処理を実行する。
- [0226] ファイル A 312 に格納される全てのパケットについて処理が終了したと判定された場合、データ解凍／並び替え処理部 305 は、処理を終了する。
- [0227] 図 15 は、本実施形態におけるストリームデータ処理部 306 の構成例を示す説明図である。
- [0228] ストリームデータ処理部 306 は、入力制御部 500、入力キュー 501

、ストリームデータ処理エンジン502、ユーザ定義関数503、出力キュー504及び出力制御部505を備える。

[0229] 入力制御部500は、入力制御を実行する。具体的には、入力制御部500は、ファイルB313から入力されるデータを受信し、受信したデータを入力キュー501に出力する。入力キュー501は、入力制御部500に入力されたデータを格納する。

[0230] ストリームデータ処理エンジン502は、入力キュー501からデータを取り出し、CQL311に基づいて取り出された情報を分析し、分析結果を出力キュー504に出力する。

[0231] ユーザ定義関数503は、CQL311において使用される関数の演算方法の定義を格納する。なお、ユーザ定義関数503の詳細については、図16を用いて後述する。

[0232] 出力キュー504は、ストリームデータ処理エンジン502から入力された分析結果を格納する。

[0233] 出力制御部505は、出力制御を実行する。具体的には、出力制御部505は、出力キュー504から分析結果を読み出し、ファイルCに分析結果を出力する。

[0234] 本実施形態では、まず、ファイルB313に格納される`valn3412`が「`$inum`」であるデータ、すなわち、「`$inum`」個のセンサ100についての値が格納されたデータがファイルB313からストリームデータ処理部306に入力される。ストリームデータ処理部306は、CQL311に基づいて処理を実行した結果をファイルC314に出力する。

[0235] 具体的には、当該ファイルC314に格納されるデータの`valn3412`には「`$snum+1`」が格納され、データ3413のうち`val[0]`～`val[$snum-1]`には、クエリ155の`select__items`句で指定された処理結果が格納され、データ3413のうち`val[$snum]`には、クエリ155の`where__condition`句で指定された条件に対する処理結果が格納される。

- [0236] ここで、`where__condition`句において指定された条件に一致する場合、`val [$snum]` には「1」が格納される。また、`where__condition`句において指定された条件に一致しない場合、`val [$snum]` には「0」が格納される。`val [$snum]` に格納される値は、後述する切り出し処理部307が切り出し範囲552（図19参照）を決定するためのフラグとして用いられる。
- [0237] 本実施形態では、`select__items`句で指定された「`$snum`」個の処理結果に、`where__condition`句の処理結果が一つ追加されるため、ファイルC314に格納されるデータのデータ3413には、「`$snum+1`」個の値が含まれる。
- [0238] 図16は、本実施形態のユーザ定義関数503の一例を示す説明図である。
- [0239] ユーザ定義関数503は、識別番号5031、関数名5032、動作説明5033及び備考5034を含む。
- [0240] 識別番号5031は、ユーザ定義関数503に含まれる関数を一意に識別するための識別子である。関数名5032は、CQL311に含まれる関数の名称である。動作説明5033は、関数名5032に対応する関数の演算内容である。備考5034は、関数名5032に対応する関数についての付加的情報である。
- [0241] 図17に示す例では、識別番号5031が「1」～「7」の関数は、「LABEL」と「num」とが動作説明5033に示す関係を満たすデータを抽出する関数である。識別番号5031が「8」及び「9」は、「LABEL」が動作説明5033に示す所定の区間内のデータを抽出する関数である。識別番号5031が「10」～「12」は、動作説明5033に示す論理演算を実行する関数である。
- [0242] なお、図8Bに示すCQL例1（353）の`GreaterThanOrEqualTo`関数では、「LABEL」は「L1」であり、「num」は「70」である。CQL例1（353）の`PositiveThreshold`

d関数では、「LABEL」は「L3」であり、「num1」及び「num2」は「300」及び「200」である。

[0243] 以下、識別番号5031が「8」及び「9」の関数によって抽出された所定の区間のデータについて説明する。

[0244] 図17は、本実施形態のユーザ定義関数503に含まれる関数によって抽出された所定の区間のデータの一例を示す説明図である。

[0245] PositiveThresholdの例は、識別番号5031が「8」の関数によって抽出された所定の区間のデータを示す。図17の動作説明5033に示すように、PositiveThresholdでは、「LABEL」の値が「num1」以上になり、「LABEL」の値が「num2」より小さくなるまでの区間のデータが抽出される。

[0246] NegativeThresholdの例は、識別番号5031が「9」の関数によって抽出された所定の区間のデータを示す。図17の動作説明5033に示すように、NegativeThresholdでは、「LABEL」が「num1」以下になり、「LABEL」が「num2」より大きくなるまでの区間のデータが抽出される。

[0247] 以下、ストリームデータ処理部306が実行する処理について説明する。

[0248] 図27は、本実施形態のストリームデータ処理部306によって実行される処理の一例を説明するフローチャートである。

[0249] ストリームデータ処理部306は、CQLを読み出し（S900）、更に変数runningを「1」に設定する（S901）。

[0250] 次に、ストリームデータ処理部306は、入力制御スレッド及び出力制御スレッドを起動する（S902、S903）。これによって、入力制御部500及び出力制御部505が、処理を開始する。入力制御部500によって実行される処理については、図28を用いて後述する。また、出力制御部505によって実行される処理については、図29を用いて後述する。

[0251] ストリームデータ処理部306は、出力キュー504に空きがあるか否かを判定する（S904）。

- [0252] 出力キュー５０４に空きがないと判定された場合、ストリームデータ処理部３０６は、出力キュー５０４に空きができるまで待つ続ける（Ｓ９０５）。
- [0253] 出力キュー５０４に空きがあると判定された場合、ストリームデータ処理部３０６は、入力キュー５０１にデータが格納されているか否かを判定する（Ｓ９０６）。
- [0254] 入力キュー５０１にデータが格納されていると判定された場合、ストリームデータ処理部３０６は、入力キュー５０１からデータを取得する（Ｓ９０７）。
- [0255] ストリームデータ処理部３０６は、ＣＱＬに基づいて取得されたデータを処理する（Ｓ９０８）。具体的には、ストリームデータ処理エンジン５０２が、ＣＱＬを実行する。
- [0256] ストリームデータ処理部３０６は、ＣＱＬの実行結果を出力キュー５０４に格納し（Ｓ９０９）、Ｓ９０４に戻り同様の処理を実行する。
- [0257] Ｓ９０６において、入力キュー５０１にデータが格納されていないと判定された場合、ストリームデータ処理部３０６は、変数 `running` が「１」であるか否かを判定する（Ｓ９１１）。
- [0258] 変数 `running` が「１」と判定された場合、ストリームデータ処理部３０６は、処理待ちの待ち状態となって（Ｓ９１０）、Ｓ９０６に戻り同様の処理を実行する。
- [0259] 変数 `running` が「１」でないと判定された場合、ストリームデータ処理部３０６は、出力キュー５０４にデータが格納されているか否かを判定する（Ｓ９１２）。
- [0260] 出力キュー５０４にデータが格納されていないと判定された場合、ストリームデータ処理部３０６は、出力キュー５０４にデータが格納されるまで待ち続ける（Ｓ９１３）。
- [0261] 出力キュー５０４にデータが格納されていると判定された場合、ストリームデータ処理部３０６は、入力制御スレッド及び出力制御スレッドを停止さ

せ（Ｓ９１４、Ｓ９１５）、処理を終了する。これによって、入力制御部５００及び出力制御部５０５の処理が終了する。

[0262] 図２８は、本実施形態の入力制御部５００によって実行される処理の一例を説明するフローチャートである。

[0263] 入力制御部５００は、入力制御の処理を開始すると（Ｓ９２０）、入力キュー５０１に空きがあるか否かを判定する（Ｓ９２２）。

[0264] 入力キュー５０１に空きがないと判定された場合、入力制御部５００は、入力キュー５０１に空きができるまで待ち続ける（Ｓ９２１）。

[0265] 入力キュー５０１に空きがあると判定された場合、入力制御部５００は、ファイルＢ３１３にデータが存在するか否かを判定する（Ｓ９２３）。

[0266] ファイルＢ３１３にデータが存在しないと判定された場合、入力制御部５００は、変数 `running` を「０」に設定して（Ｓ９２６）、処理を終了する（Ｓ９２７）。

[0267] ファイルＢ３１３にデータが存在すると判定された場合、入力制御部５００は、ファイルＢ３１３からデータを取得し（Ｓ９２４）、取得されたデータを入力キュー５０１に格納し（Ｓ９２５）、Ｓ９２２に戻り同様の処理を実行する。

[0268] 図２９は、本実施形態の出力制御部５０５によって実行される処理の一例を説明するフローチャートである。

[0269] 出力制御部５０５は、出力制御の処理を開始すると（Ｓ９３０）、出力キュー５０４にデータが存在するか否かを判定する（Ｓ９３１）。

[0270] 出力キュー５０４にデータが存在しないと判定された場合、出力制御部５０５は、出力キュー５０４にデータが格納するまで待ち続ける（Ｓ９３２）。

[0271] 出力キュー５０４にデータが存在すると判定された場合、出力制御部５０５は、出力キュー５０４からデータを取得し（Ｓ９３３）、取得されたデータをファイルＣ３１４に格納する（Ｓ９３４）。

[0272] 出力制御部５０５は、その後、Ｓ９３１に戻り同様の処理を実行する。

- [0273] 図18Aは、本実施形態の切り出し処理部307によって実行される処理の一例を説明するフローチャートである。
- [0274] 切り出し処理部307は、ストリームデータ処理部306の処理が終了した後に処理を開始する。
- [0275] 切り出し処理部307は、変数`prev_cnt`、変数`post_cnt`及び変数`latest`の初期化と、FIFOバッファの生成及び初期化と、`indata`バッファの定義及び初期化とを実行する（S570）。
- [0276] 具体的には、切り出し処理部307は、変数`prev_cnt`及び変数`post_cnt`を「0」に初期化する。また、切り出し処理部307は、変数`latest`を「`$start-$prev-1`」と初期化する。また、切り出し処理部307は、（`$prev+1`）段のFIFOバッファを生成し、生成されたFIFOバッファを「0」に初期化する。なお、変数`prev_cnt`及び変数`post_cnt`は、後述する切り出し処理で用いられる変数である。
- [0277] 切り出し処理部307は、ファイルC314からデータを一つ読み出し、`indata`に格納する（S571）。なお、ファイルC314には、`select_items`句の演算結果、1行に対応するデータ構造2（341）のデータが複数個格納されている。
- [0278] 切り出し処理部307は、ファイルC314から読み出されたデータが、ファイルC314に格納される終端のデータであるか否かを判定する（S572）。例えば、読み出されたデータの`datetime3411`が、グローバル変数309に含まれる「`$end`」と同一か否かを判定することによって可能である。
- [0279] ファイルC314から読み出されたデータが、ファイルC314に格納される終端のデータでないと判定された場合、切り出し処理部307は、`indata`に含まれる`datetime3411`及び変数`latest`を用いて、変数`diff`を「`datetime-latest`」と設定し（S573）、S575に進む。

- [0280] ファイルC 3 1 4から読み出されたデータが、ファイルC 3 1 4に格納される終端のデータであると判定された場合、切り出し処理部3 0 7は、変数 `d i f f` を「`$ p r e v + 1`」に設定し（S 5 7 4）、S 5 7 5に進む。
- [0281] 切り出し処理部3 0 7は、変数 `d i f f` に設定された値を変数 `i` に代入し、変数 `i` が「0」より小さくなるまで繰り返し処理を実行する（S 5 7 5）。
- [0282] 切り出し処理部3 0 7は、変数 `i` が「1」であるか否かを判定する（S 5 7 6）。
- [0283] 変数 `i` が「1」でないと判定された場合、切り出し処理部3 0 7は、F I F Oバッファに挿入するダミーデータ（以下、挿入データとも記載する）を生成し（S 5 7 7）、S 5 8 0に進む。
- [0284] 具体的には、データ構造2（3 4 1）に示すような挿入データが生成される。この場合、挿入データの `d a t e t i m e 3 4 1 1` には「`l a t e s t + 1`」が設定され、`v a l n 3 4 1 2` には「`$ s n u m + 1`」が設定される。また、挿入データのデータ3 4 1 3のうち `v a l [ 0 ] ~ [ $ s n u m - 1 ]` には空データを示す値（N a N）が設定され、`v a l [ $ s n u m ]` には「0」が設定される。
- [0285] 変数 `i` が「1」であると判定された場合、切り出し処理部3 0 7は、ファイルC 3 1 4から読み出されたデータ（`i n d a t e`）が、ファイルC 3 1 4に格納される終端のデータであるか否かを判定する（S 5 7 8）。当該判定は、S 5 7 2と同一の方法が用いられる。
- [0286] ファイルC 3 1 4から読み出されたデータ（`i n d a t e`）が、ファイルC 3 1 4に格納される終端のデータであると判定された場合、切り出し処理部3 0 7は、処理を終了する。
- [0287] ファイルC 3 1 4から読み出されたデータ（`i n d a t e`）が、ファイルC 3 1 4に格納される終端のデータでないと判定された場合、切り出し処理部3 0 7は、入力データを生成し（S 5 7 9）、S 5 8 0に進む。具体的には、切り出し処理部3 0 7は、`i n d a t a`に格納された値を入力データに

設定に設定する。

- [0288] 切り出し処理部307は、入力データを用いて切り出し処理を実行する（S580）。切り出し処理の詳細については、図18Bを用いて後述する。
- [0289] 切り出し処理部307は、処理を繰り返し実行するか否かを判定する（S581、S575）。すなわち、切り出し処理部307は、S575のfor文ループの実行を続行するか否かを判定する。具体的には、変数*i*をデクリメントし、「*i* > 0」であるか否かが判定される。判定結果が真の場合は、処理を繰り返し実行すると判定される。
- [0290] 処理を繰り返し実行すると判定された場合には、切り出し処理部307は、S575に戻り同様の処理（S575～S581）を実行する。
- [0291] 処理を繰り返し実行しないと判定された場合には、切り出し処理部307は、S571に戻り同様の処理（S571～S581）を実行する。
- [0292] 図18Bは、本実施形態の切り出し処理の詳細を説明するフローチャートである。
- [0293] 切り出し処理部307は、挿入データの*datetime*を、変数*latest*に設定する（S582）。
- [0294] 切り出し処理部307は、FIFOバッファに一つの挿入データを入力し、また、FIFOバッファから一つのデータを出力する（S583）。以下では、FIFOバッファから出力されるデータを出力データとも記載する。
- [0295] 切り出し処理部307は、入力された挿入データの*val [\$snum]*が「1」であるか否かを判定する（S584）。
- [0296] 入力された挿入データの*val [\$snum]*が「1」でないと判定された場合、切り出し処理部307は、S586に進む。
- [0297] 入力された挿入データの*val [\$snum]*が「1」と判定された場合、切り出し処理部307は、変数*prev_cnt*の値を「1」加算し（S585）、S586に進む。
- [0298] 切り出し処理部307は、出力された出力データの*val [\$snum]*が「1」であるか否かを判定する（S586）。

- [0299] 出力された出力データの `val [$snum]` が「1」でないと判定された場合、切り出し処理部307は、S588に進む。
- [0300] 出力された出力データの `val [$snum]` が「1」であると判定された場合、切り出し処理部307は、変数 `prev_cnt` の値を「1」減算し（S587）、S588に進む。
- [0301] 切り出し処理部307は、変数 `prev_cnt`、変数 `post_cnt` 及び入力された挿入データの `val [$snum]` の値に基づいて、切り出し条件判定処理を実行する（S588）。具体的には以下に示す判定処理が実行される。
- [0302] S588では、切り出し処理部307は、3つの判定処理を実行する。
- [0303] 第1に、切り出し処理部307は、変数 `prev_cnt` の値が「0」より大きいか否かを判定する。以下、当該判定を切り出し条件（1）と記載する。
- [0304] 第2に、切り出し処理部307は、出力された出力データの `val [$snum]` の値が「1」であるか否かを判定する。以下、当該判定を切り出し条件（2）と記載する。
- [0305] 第3に、切り出し処理部307は、変数 `post_cnt` の値が「0」より大きいか否かを判定する。以下、当該判定を切り出し条件（3）と記載する。
- [0306] 切り出し処理部307は、切り出し条件判定処理の実行結果に基づいて、切り出し条件（1）、（2）又は（3）の少なくともいずれかを満たすか否かを判定する（S589）。
- [0307] 切り出し条件（1）、（2）又は（3）のいずれも満たさないと判定された場合、切り出し処理部307は、切り出し処理S580の処理を終え、S581へ進む。
- [0308] 切り出し条件（1）、（2）又は（3）の少なくともいずれかを満たすと判定された場合、切り出し処理部307は、切り出し条件判定処理の実行結果に基づいて、切り出し条件（1）又は（2）の少なくともいずれかを満た

すか否かを判定する（S590）。

[0309] 切り出し条件（１）又は（２）の少なくともいずれかを満たすと判定された場合、切り出し処理部３０７は、「\$post」の値を変数post_cntに設定し（S591）、S593に進む。

[0310] 切り出し条件（１）又は（２）のいずれも満たさないと判定された場合、切り出し処理部３０７は、変数post_cntを「１」減算し（S592）、S593に進む。

[0311] 切り出し処理部３０７は、foutデータをファイルD315に出力し（S593）、切り出し処理S580の処理を終え、S581へ進む。

[0312] 当該処理によって、切り出し条件を満たすもののみがファイルD315に出力される。なお、ファイルD315には、クエリ155のwhere_condition句で指定された条件を満たすデータを含む前後のオフセットが付加された出力結果が格納される。

[0313] 具体的には、切り出し処理部３０７は、S588、S589において、切り出し条件（１）を満たすこと示すフラグ、切り出し条件（２）を満たすことを示すフラグ、及び切り出し条件（３）を満たすことを満たすフラグを各挿入データに付与する。さらに、切り出し処理部３０７は、当該フラグに基づいて各切り出し条件の判定を行う。

[0314] 以下、当該切り出し処理部３０７が実行する処理によって出力される出力結果について説明する。

[0315] 図１９は、本実施形態の切り出し処理部３０７によって出力される出力結果の一例を示す説明図である。

[0316] クエリ例３（５５０）が入力された場合に、切り出し処理部３０７は、出力結果５５１に示すような結果が出力される。

[0317] 具体的には、where_condition句においてLABELが「L1」であるSensor1の値が「100」以上である条件成立範囲５５４、前方オフセット５５３及び後方オフセット５５５を含む切り出し範囲５５２、又は、条件成立範囲５５４、前方オフセット５５３、後方オフセット

- ５５５及び結合オフセット５５６を含む切り出し範囲５５２が出力される。
- [0318] すなわち、切り出し範囲５５２に示すような範囲のデータが切り出され、出力結果として出力される。
- [0319] なお、結合オフセット５５６は、前方オフセット５５３と後方オフセット５５５とが重なる部分を一つのオフセットとして結合したものである。
- [0320] 当該出力結果は、図１８のＳ５８９において付与されたフラグに基づいて生成されたものである。すなわち、切り出し条件（１）～（３）のいずれかのフラグが付与された挿入データが切り出し範囲５５２と出力される。具体的には以下の通りである。
- [0321] 切り出し条件（２）のフラグが付与された挿入データは、切り出し範囲５５２のうち条件成立範囲５５４に含まれるデータとして出力される。
- [0322] 切り出し条件（１）のフラグが付与された挿入データは、切り出し範囲５５２のうち前方オフセット５５３に含まれるデータとして出力される。
- [0323] 切り出し条件（３）のフラグが付与された挿入データは、切り出し範囲５５２のうち後方オフセット５５５に含まれるデータとして出力される。
- [0324] 切り出し処理部３０７は、それぞれの切り出し条件（１）～（３）に一致する挿入データを切り出し、切り出されたデータを時間順に並び替えて（ソートして）、出力する。
- [0325] なお、切り出し条件（１）～（３）のいずれかのフラグが２以上付与された挿入データは、結合オフセット５５６を含む切り出し範囲５５２として出力される。
- [0326] 図２０は、本発明の実施形態のデータ出力部３０８によって実行される処理の一例を説明するフローチャートである。
- [0327] データ出力部３０８は、tmpバッファを定義する（Ｓ６０５）。
- [0328] データ出力部３０８は、ファイルＤ３１５からデータを読み出し、読み出されたデータをtmpに格納する（Ｓ６００）。読み出されるデータは、select__items句の演算結果、１行に対応するデータ構造２（３４１）に示すようなデータである。

- [0329] データ出力部308は、読み出されたデータの`datetime3411`を参照し、検索結果抽出処理を実行する(S601)。具体的には、以下の2つの処理が実行される。
- [0330] 第1に、データ出力部308は、`datetime3411`に格納される値が、クエリ155の`where__timerange`句において指定された時間範囲に含まれるか否かを判定する。すなわち、`datetime3411`に格納される値が、「\$start」以上かつ「\$end」以下であるか否かを判定する。以下、当該判定を出力条件(1)と記載する。
- [0331] 出力条件(1)を満たすか否かの判定処理が実行される理由は、SQL310によって検索される範囲のデータが、クエリ155において指定された時間範囲より広い範囲のデータが対象となっているためである。すなわち、図11Bにおいて説明したように、SQL310を生成するときに用いられる「\$db__start」(図11B参照)は、「\$start」より前の時刻のデータについても処理しているためである。
- [0332] 第2に、データ出力部308は、間引き処理を実行する。具体的には、`datetime3411`に格納される値を、クエリ155の`step`句において指定された値、すなわち、「\$step」で除算した余りが「0」となるか否かを判定する。以下、当該判定を出力条件(2)と記載する。クエリ例1(350)に示す例では、出力条件(2)によって、5秒ごとのデータが出力される。
- [0333] データ出力部308は、S601における検索結果抽出処理の結果に基づいて、出力条件(1)及び出力条件(2)を満たすか否かを判定する(S602)。
- [0334] 出力条件(1)及び出力条件(2)の両判定を同時に満たさないと判定された場合、データ出力部308は、S604に進む。すなわち、当該データは出力されない。
- [0335] 出力条件(1)及び出力条件(2)を満たすと判定された場合、データ出力部308は、検索結果生成処理を実行する(S603)。具体的には、デ

ータ出力部308は、tmpに格納されるデータをデータ構造3(342)に示すようなCSV形式に変換して検索結果ファイル156に出力する。

[0336] データ出力部308は、ファイルD315に格納される全てのデータについて処理が完了したか否かを判定する(S604)。

[0337] 本実施形態では、データベース検索処理では1時間単位に圧縮された圧縮データを検索し、データ解凍/並び替え処理では1時間単位に圧縮された圧縮データを解凍して1秒単位のデータを生成し、以後、ストリームデータ処理、切り出し処理、出力データ生成処理では1秒単位のデータについて処理を実行する。なお、出力データ生成処理では、1秒単位でデータを処理するが、\$stepで指定した単位でデータを出力する。

[0338] これによって、プラント102において異常が発生した場合に、当該異常の検索精度を落とすことなく、高速に検索することができ、また、異常発生の原因を特定するためのデータ範囲を出力することが可能となる。

[0339] 図21は、本実施形態において、特徴値検索処理を実行する場合のクエリ155及びSQL310の一例を示す説明図である。

[0340] 特徴値検索処理を実行する場合、クエリ155にmeta__search句が追加される点が異なる。これによって、DATテーブル208に含まれるMaxValカラム2084及びMinValカラム2085を用いた検索が可能となる。

[0341] 図21に示すようにクエリ例4(700)には、MaxValカラム2084の値が「100」より大きいデータが検索の対象となる。

[0342] データ検索モジュール157は、meta__search句が含まれるクエリ155を受け付けた場合、特徴値検索用SQL生成部302は、図21に示すようなSQL例4-1(701)を生成し、SQL例4-1(701)が実行された後に、SQL例4-2(702)を生成する。

[0343] SQL例4-1(701)は、時刻切り出し用SQLであり、特徴値に一致するデータの時刻(Date time)を抽出するためのSQLである。

[0344] SQL例4-2(702)は、特徴値検索用SQLであり、SQL例4-

1 (701) によって抽出された `Date time` の範囲において、データベース 154 の検索を実行するための SQL である。

[0345] SQL 例 4-2 (702) は、SQL 例 1 (352) と同様の SQL 310 であるが、SQL 例 1 (352) と比較して、SQL 例 4-2 (702) では SQL 例 4-1 (701) の実行によって検索範囲が限定されるため、検索処理の時間を大幅に短縮することが可能となる。

[0346] 例えば、異常値があらかじめわかっている場合は、当該異常値を用いて検索処理を実行することによって、全検索する場合に比べて検索するデータの範囲を絞り込むことが可能になるため、処理時間を短縮することができる。

[0347] また、図 21 のクエリ例 4 において、ユーザは必ずしも `meta__search` 句を指定する必要はない。一般に閾値判定において、ある閾値を超過した時系列データは、最大値が当該閾値を超過したパケット集合に含まれている。また同様に、ある閾値を下回る時系列データは、最小値が当該閾値を下回るパケット集合に含まれている。このようなクエリ変換規則を管理することにより、`meta__search` 句を自動生成することができる。具体的には、図 21 のクエリ例 4 において、`where__condition` 句において `greater than` 句と閾値が指定されている場合、`meta__search` 句において最大値が閾値を超過する句として `Max Val > 閾値` を自動付与する。また `where__condition` 句において `less than` 句と閾値が指定されている場合、`meta__search` 句において最大値が閾値を超過する句として `Min Val < 閾値` を自動付与する。このようなクエリ変換規則は、システムにおいてあらかじめ登録しておいてもよいし、ユーザが後で登録可能にしてもよい。

[0348] 図 22 は、本実施形態において、特徴値検索用の SQL 310 を生成するための SQL テンプレートを示す説明図である。

[0349] 特徴値検索用 SQL 生成部 302 は、時刻切り出し用 SQL テンプレート 710 及び特徴値検索用 SQL テンプレート 711 を保持する。

[0350] 時刻切り出し用 SQL テンプレート 710 は、特徴値検索を実行する場合

のSQL 310を生成するためのテンプレートであり、特徴値に一致するデータが含まれる圧縮データの時刻を抽出するためのSQLである。

[0351] 具体的には、IDカラム2072とIDカラム2082とが同一であるSIDテーブル207とDATテーブル208とを組み合わせたテーブルから、時刻(Dat et ime)を所定の条件に基づいて検索するSQLであることを示す。

[0352] 所定の条件は、where句以下に定義されている。すなわち、データを時間順に並び替え、指定されたセンサ名、指定された特徴値及び指定された時間区間に一致するデータを抽出することが定義されている。

[0353] 特徴値検索用SQLテンプレート711は、特徴値検索を実行する場合のSQL 310を生成するためのテンプレートであり、時刻切り出し用SQLテンプレート710を用いて生成されたSQLによって抽出された時刻範囲において、圧縮データを検索するためのSQLである。

[0354] 図22に示す例では、IDカラム2072とIDカラム2082とが同一であるSIDテーブル207とDATテーブル208とを組み合わせたテーブルから、時刻(Dat et ime)、センサ名(SID. Name)及び圧縮データ(DAT. CDATA)を所定の条件に基づいて検索するSQLであることを示す。

[0355] 所定の条件は、where句以下に定義される。すなわち、データを時間順に並び替え、指定されたセンサ名及び指定された時間区間に一致するデータを取得することが定義されている。

[0356] 図23は、本実施形態の特徴値検索用SQL生成部302によって実行される処理の一例を説明するフローチャートである。

[0357] 特徴値検索用SQL生成部302は、グローバル変数309を用いて、「\$meta__signal」及び「\$meta__cond」を生成する(S720)。具体的には、特徴値検索用SQL生成部302は、グローバル変数309に含まれる「\$meta」を用いて、\$meta分解処理730を実行する。すなわち、特徴値検索用SQL生成部302は、「\$meta」

を「\$meta__signal」及び「\$meta__cond」に分解する。

[0358] 特徴値検索用SQL生成部302は、ワーク用の変数\$d_t__tmpを定義する（S7201）。

[0359] 特徴値検索用SQL生成部302は、時刻切り出し用SQLを生成する（S721）。

[0360] 具体的には、まず、特徴値検索用SQL生成部302は、「\$meta__start」=「\$start」と設定し、さらに、分及び秒単位の値を「0」に初期化する。また、特徴値検索用SQL生成部302は、「\$meta__end」=「\$end」と設定する。次に、特徴値検索用SQL生成部302は、「\$meta__signal」、「\$meta__cond」、「\$meta__start」及び「\$meta__end」を時刻切り出し用SQLテンプレート710に代入して、時刻切り出し用SQLを生成する。

[0361] 特徴値検索用SQL生成部302は、時刻切り出し用SQLをデータベース154に対して発行する（S722）。

[0362] 特徴値検索用SQL生成部302は、時刻切り出し用SQLを実行したデータベース154から実行結果を取得し、\$d_t__tmpへ格納する（S723）。具体的には、時刻切り出し用SQLにおいて指定された条件に一致する時刻データが実行結果として取得され、\$d_t__tmpへ格納される。

[0363] 特徴値検索用SQL生成部302は、\$d_t__tmpに格納された1行分のデータがNullである否かを判定する（S724）。すなわち、全ての実行結果について処理が終了したか否かが判定される。

[0364] \$d_t__tmpに格納された1行分のデータがNullであると判定された場合、特徴値検索用SQL生成部302は、\$db__input=\$inputとして\$db__inputを生成し、さらに、特徴値検索用SQLを生成し（S725）、処理を終了する。具体的には、特徴値検索用SQL生成部302は、特徴値検索用SQLテンプレート711に「\$db__input」及び「\$db__dtset」を代入することによって特徴値検索用S

QLを生成する。

[0365] \$dt__tmpに格納された1行分のデータがNullでないと判定された場合、特徴値検索用SQL生成部302は、時刻\$dt__tmpのセンサデータが格納されている、時系列データブロックの開始時刻、及び、終了時刻、すなわち、「\$db__start」及び「\$db__end」を算出する（S726）。

[0366] 具体的には、「\$db__start」については、特徴値検索用SQL生成部302は「\$db__start=\$dt__tmp-\$range」を計算し、さらに、当該計算結果の分及び秒単位の値を「0」に初期化される。また、「\$db__end」については、「\$db__end=\$db__start+3599」を計算した値が格納される。「3599」を加算する理由は、本実施形態では1時間単位でデータを圧縮してデータベースへ格納しているためである。

[0367] 特徴値検索用SQL生成部302は、S726で算出された「\$db__start」及び「\$db__end」を用いて、「\$db__dtset」を生成し（S727）、S723に戻り同様の処理を実行する。具体的には、特徴値検索用SQL生成部302は、「\$db__start」及び「\$db__end」を用いて、\$db__dtset生成処理を実行する。

[0368] 図24は、本実施形態において、データ検索モジュール157によって実行される処理を並列に実行する場合を説明するフローチャートである。

[0369] CPU121は、複数のコアを備えており、図10に示すように、DB検索3204の実行と同時に、各プロセス1～4においてそれぞれの処理を実行することも可能である。

[0370] S3200～S3202は、図9のS320～S323及びS327と同一の処理であるため説明を省略する。

[0371] データ検索モジュール157は、プロセス1～4を起動し（S3203）、各プロセスでそれぞれの処理を実行する（S3204～S3208）。

[0372] 具体的には、プロセス1ではデータ解凍／並び替え処理が実行され、プロ

セス2ではストリームデータ処理が実行され、プロセス3では切り出し処理が実行され、また、プロセス4では出力データ生成処理が実行される。

[0373] これによって、各処理の結果をファイルA312～ファイルD315に出力するのではなく、一つのプロセスの出力結果を他のプロセスに受け渡すことによって高速に処理することが可能となる。

[0374] 本実施形態によれば、時系列データを所定の時間単位に圧縮した圧縮データをデータベースに格納することによって、データベースに格納されるデータ量を削減できる。また、一つのクエリを受け付けることによって、圧縮データ検索のためのSQLと、解凍されたデータを処理するCQLとを生成できる。さらに、CQLにおいて指定された条件を満たす範囲と、当該範囲の前後の区間とをあわせた切り出し範囲を出力することによって、異常が発生した箇所だけではなく、異常が発生した前後の時系列データも出力することができ、異常箇所及び異常原因の特定のための追跡が容易になる。

[0375] これによって、センタサーバ104のディスク容量の増加を抑制するとともに、プラントにおける異常原因を解析することが容易となる。

[0376] さらに、特徴値とともに圧縮データをデータベース154に格納し、当該特徴値を用いて検索を実行することによって、処理時間を削減することが可能となる。

[0377] 以上、本発明を添付の図面を参照して詳細に説明したが、本発明はこのような具体的構成に限定されるものではなく、添付した請求の範囲の趣旨内における様々な変更及び同等の構成を含むものである。

産業上の利用可能性

[0378] 本発明は、膨大なデータを扱う工場やプラントにおける設備等の予防保全に適用することができる。

請求の範囲

[請求項1]

プロセッサと、前記プロセッサに接続されるメモリとを備え、データベースを管理する計算機におけるデータベース管理方法であって、

前記データベースは、所定の条件に基づいて圧縮された複数の圧縮データを格納し、

前記方法は、

前記計算機に接続されるクライアント計算機から前記データベースへのクエリを受け付けた場合に、前記計算機が、前記受け付けたクエリを解析する第1のステップと、

前記計算機が、前記受け付けたクエリの解析結果に基づいて、前記データベースから一つ以上の前記圧縮データを検索するための第1の問い合わせを生成する第2のステップと、

前記計算機が、前記受け付けたクエリの解析結果に基づいて、前記第1の問い合わせの応答結果である前記一つ以上の圧縮データから取得される複数の時系列データに対する検索を実行するための第2の問い合わせを生成する第3のステップと、

前記計算機が、前記データベースに前記第1の問い合わせを発行して、前記データベースから前記第1の問い合わせの応答結果として一つ以上の前記圧縮データを取得する第4のステップと、

前記計算機が、前記第1の問い合わせに対する応答結果として取得された一つ以上の圧縮データを解凍することによって前記複数の時系列データを取得する第5のステップと、

前記計算機が、前記取得された複数の時系列データに対して前記第2の問い合わせを実行する第6のステップと、

前記計算機が、前記第2の問い合わせに対する応答結果に基づいて、前記取得された複数の時系列データから所定のデータを抽出する第7のステップと、

前記計算機が、前記第7のステップにおいて抽出された所定のデー

タから前記クライアント計算機に出力するためのデータを抽出し、出力結果を生成する第 8 のステップと、を含むことを特徴とするデータベース管理方法。

[請求項2] 前記圧縮データは、前記複数の時系列データが所定の時間単位に圧縮された圧縮データであり、

前記第 2 の問い合わせは、前記複数の時系列データから所定の閾値の条件を満たす第 1 のデータ範囲を検索するための問い合わせであり、

前記第 6 のステップは、前記計算機が、前記第 1 のデータ範囲に含まれる前記時系列データにフラグを付与し、

前記第 7 のステップは、前記計算機が、前記時系列データに付与されたフラグに基づいて、前記第 1 のデータ範囲を含む第 2 のデータ範囲を抽出することを特徴とする請求項 1 に記載のデータベース管理方法。

[請求項3] 前記第 2 のデータ範囲は、

前記第 1 のデータ範囲の時刻が最も過去の前記時系列データから第 1 の時間さかのぼった範囲に含まれる前記時系列データを抽出する第 3 のデータ範囲と、

前記第 1 のデータ範囲の時刻が最新の前記時系列データから第 2 の時間進んだ範囲に含まれる前記時系列データを抽出する第 4 のデータ範囲と、を含むことを特徴とする請求項 2 に記載のデータベース管理方法。

[請求項4] 前記第 8 のステップは、前記第 2 のデータ範囲に含まれる時系列データのうち、所定の時間間隔の前記時系列データを抽出するステップを含むことを特徴とする請求項 2 に記載のデータベース管理方法。

[請求項5] 前記データベースは、検索キーとなる特徴情報を付加した前記圧縮データを格納し、

前記第 1 の問い合わせは、

前記特徴情報を含み、

当該第 1 の問い合わせに含まれる特徴情報に一致する前記特徴情報が付加された前記圧縮データを検索するための問い合わせであることを特徴とする請求項 1 に記載のデータベース管理方法。

[請求項6]

前記特徴情報は、前記圧縮データが圧縮される前の複数の時系列データにおける最大値、最小値、平均値、分散値又は周波数スペクトル上の特定点の少なくともいずれかであることを特徴とする請求項 5 に記載のデータベース管理方法。

[請求項7]

前記第 1 の問い合わせは、前記データベースから前記圧縮データを検索するための SQL であり、

前記第 2 の問い合わせは、前記時系列データから所定のデータを抽出するための CQL であることを特徴とする請求項 1 に記載のデータベース管理方法。

[請求項8]

プロセッサと、前記プロセッサに接続されるメモリとを備え、データベースを管理する計算機であって、

前記データベースは、所定の条件に基づいて圧縮された複数の圧縮データを格納し、

前記計算機は、

前記計算機に接続されるクライアント計算機から前記データベースへのクエリを受け付けた場合に、前記受け付けたクエリを解析し、

前記受け付けたクエリの解析結果に基づいて、前記データベースから一つ以上の前記圧縮データを検索するための第 1 の問い合わせを生成し、

前記受け付けたクエリの解析結果に基づいて、前記第 1 の問い合わせの応答結果である前記一つ以上の圧縮データから取得される複数の時系列データに対する検索を実行するための第 2 の問い合わせを生成し、

前記データベースに前記第 1 の問い合わせを発行して、前記データ

ベースから前記第 1 の問い合わせの応答結果として一つ以上の前記圧縮データを取得し、

前記第 1 の問い合わせに対する応答結果として取得された一つ以上の圧縮データを解凍することによって前記複数の時系列データを取得し、

前記取得された複数の時系列データに対して前記第 2 の問い合わせを実行し、

前記第 2 の問い合わせに対する応答結果に基づいて、前記取得された複数の時系列データから所定のデータを抽出し、

前記抽出された所定のデータから前記クライアント計算機に出力するためのデータを抽出し、出力結果を生成することを特徴とする計算機。

[請求項9]

前記圧縮データは、前記複数の時系列データが所定の時間単位に圧縮された圧縮データであり、

前記第 2 の問い合わせは、前記複数の時系列データから所定の閾値の条件を満たす第 1 のデータ範囲を検索するための問い合わせであり、

前記計算機は、

前記取得された複数の時系列データに対して前記第 2 の問い合わせを実行する場合に、前記第 1 のデータ範囲に含まれる前記時系列データにフラグを付与し、

前記第 2 の問い合わせに対する応答結果に基づいて、前記取得された複数の時系列データから所定のデータを抽出する場合に、前記時系列データに付与されたフラグに基づいて、前記第 1 のデータ範囲を含む第 2 のデータ範囲を抽出することを特徴とする請求項 8 に記載の計算機。

[請求項10]

前記第 2 のデータ範囲は、

前記第 1 のデータ範囲の時刻が最も過去の前記時系列データから第

１の時間さかのぼった範囲に含まれる前記時系列データを抽出する第３のデータ範囲と、

前記第１のデータ範囲の時刻が最新の前記時系列データから第２の時間進んだ範囲に含まれる前記時系列データを抽出する第４のデータ範囲と、を含むことを特徴とする請求項９に記載の計算機。

[請求項11] 前記データベースは、検索キーとなる特徴情報を付加した前記圧縮データを格納し、

前記第１の問い合わせは、

前記特徴情報を含み、

当該第１の問い合わせに含まれる特徴情報に一致する前記特徴情報が付加された前記圧縮データを検索するための問い合わせであることを特徴とする請求項８に記載の計算機。

[請求項12] 前記特徴情報は、前記圧縮データが圧縮される前の複数の時系列データにおける最大値、最小値、平均値、又は分散値の少なくともいずれかであることを特徴とする請求項１１に記載の計算機。

[請求項13] 前記第１の問い合わせは、前記データベースから前記圧縮データを検索するためのＳＱＬであり、

前記第２の問い合わせは、前記時系列データから所定のデータを抽出するためのＣＱＬであることを特徴とする請求項８に記載の計算機。

[請求項14] 観測対象の物理量を測定するセンサと、前記センサが測定した前記物理量を時系列データとして収集する計算機と、前記第１の計算機から前記時系列データをデータベースに格納する管理サーバと、を備えるセンサノードシステムにおいて、

前記計算機は、第１のプロセッサと、前記第１のプロセッサに接続される第１のメモリとを備え、

前記管理サーバは、第２のプロセッサと、前記第２のプロセッサに接続される第２のメモリとを備え、

前記計算機と前記管理サーバとはネットワークを介して接続され、
前記データベースは、所定の条件に基づいて前記時系列データが圧縮された複数の圧縮データを格納し、

前記管理サーバは、

前記管理サーバに前記ネットワークを介して接続されるクライアント計算機から前記データベースへのクエリを受け付けた場合に、前記受け付けたクエリを解析し、

前記受け付けたクエリの解析結果に基づいて、前記データベースから一つ以上の前記圧縮データを検索するための第1の問い合わせを生成し、

前記受け付けたクエリの解析結果に基づいて、前記第1の問い合わせの応答結果である前記一つ以上の圧縮データから取得される複数の前記時系列データに対する検索を実行するための第2の問い合わせを生成し、

前記データベースに前記第1の問い合わせを発行して、前記データベースから前記第1の問い合わせの応答結果として一つ以上の前記圧縮データを取得し、

前記第1の問い合わせに対する応答結果として取得された一つ以上の圧縮データを解凍することによって前記複数の時系列データを取得し、

前記取得された複数の時系列データに対して前記第2の問い合わせを実行し、

前記第2の問い合わせに対する応答結果に基づいて、前記取得された複数の時系列データから所定のデータを抽出し、

前記抽出された所定のデータから前記クライアント計算機に出力するためのデータを抽出し、出力結果を生成することを特徴とするセンサネットワークシステム。

[請求項15]

前記圧縮データは、前記複数の時系列データが所定の時間単位に圧

縮された圧縮データであり、

前記第 2 の問い合わせは、前記複数の時系列データから所定の閾値の条件を満たす第 1 のデータ範囲を検索するための問い合わせであり、

、

前記管理サーバは、

前記取得された複数の時系列データに対して前記第 2 の問い合わせを実行する場合に、前記第 1 のデータ範囲に含まれる前記時系列データにフラグを付与し、

前記第 2 の問い合わせに対する応答結果に基づいて、前記取得された複数の時系列データから所定のデータを抽出する場合に、前記時系列データに付与されたフラグに基づいて、前記第 1 のデータ範囲を含む第 2 のデータ範囲を抽出することを特徴とする請求項 14 に記載のセンサネットワークシステム。

[請求項16]

前記第 2 のデータ範囲は、

前記第 1 のデータ範囲の時刻が最も過去の前記時系列データから第 1 の時間さかのぼった範囲に含まれる前記時系列データを抽出する第 3 のデータ範囲と、

前記第 1 のデータ範囲の時刻が最新の前記時系列データから第 2 の時間進んだ範囲に含まれる前記時系列データを抽出する第 4 のデータ範囲と、を含むことを特徴とする請求項 15 に記載のセンサネットワークシステム。

[請求項17]

前記データベースは、検索キーとなる特徴情報を付加した前記圧縮データを格納し、

前記第 1 の問い合わせは、

前記特徴情報を含み、

当該第 1 の問い合わせに含まれる特徴情報に一致する前記特徴情報が付加された前記圧縮データを検索するための問い合わせであることを特徴とする請求項 14 に記載のセンサネットワークシステム。

[請求項18] 前記特徴情報は、前記圧縮データが圧縮される前の複数の時系列データにおける最大値、最小値、平均値、又は分散値の少なくともいずれかであることを特徴とする請求項17に記載のセンサネットワークシステム。

[請求項19] 前記第1の問い合わせは、前記データベースから前記圧縮データを検索するためのSQLであり、

前記第2の問い合わせは、前記時系列データから所定のデータを抽出するためのCQLであることを特徴とする請求項14に記載のセンサネットワークシステム。

[請求項20] プロセッサと、前記プロセッサに接続されるメモリとを備え、データベースを管理する計算機におけるデータベース検索プログラムであって、

前記データベースは、所定の条件に基づいて圧縮された複数の圧縮データを格納し、

前記プログラムは、

前記計算機に接続されるクライアント計算機から前記データベースへのクエリを受け付けた場合に、前記受け付けたクエリを解析する手順と、

前記受け付けたクエリの解析結果に基づいて、前記データベースから一つ以上の前記圧縮データを検索するための第1の問い合わせを生成する手順と、

前記受け付けたクエリの解析結果に基づいて、前記第1の問い合わせの応答結果である前記一つ以上の圧縮データから取得される複数の時系列データに対する検索を実行するための第2の問い合わせを生成する手順と、

前記データベースに前記第1の問い合わせを発行して、前記データベースから前記第1の問い合わせの応答結果として一つ以上の前記圧縮データを取得する手順と、

前記第 1 の問い合わせに対する応答結果として取得された前記一つ以上の圧縮データを解凍することによって前記複数の時系列データを取得する手順と、

前記取得された複数の時系列データに対して前記第 2 の問い合わせを実行する手順と、

前記第 2 の問い合わせに対する応答結果に基づいて、前記取得された複数の時系列データから所定のデータを抽出する手順と、

前記抽出された所定のデータから前記クライアント計算機に出力するためのデータを抽出し、出力結果を生成する手順と、を前記計算機に実行させることを特徴とするデータベース検索プログラム。

[請求項21]

時系列データに対して時系列処理を行うことで特徴量を生成し、時系列を時系列ブロックに分割し、該特徴量を検索キーとなるメタ情報として時系列ブロックとともに時系列蓄積装置に格納する時系列データ格納手段と、

ユーザからの問い合わせに応じて該特徴量を検索キーに時系列ブロックを検索する第一の問合せと、時系列ブロック内の時系列を検索する第二の問い合わせを生成する手段と、

第一の問い合わせにより該時系列ブロックを検索する第一の検索手段と、

得られた時系列ブロックを逐次第二の問い合わせに基づき時系列処理を行って時系列を検索し、問合せ結果を出力する第二の検索手段と、を有することを特徴とする時系列データ管理方法。

[請求項22]

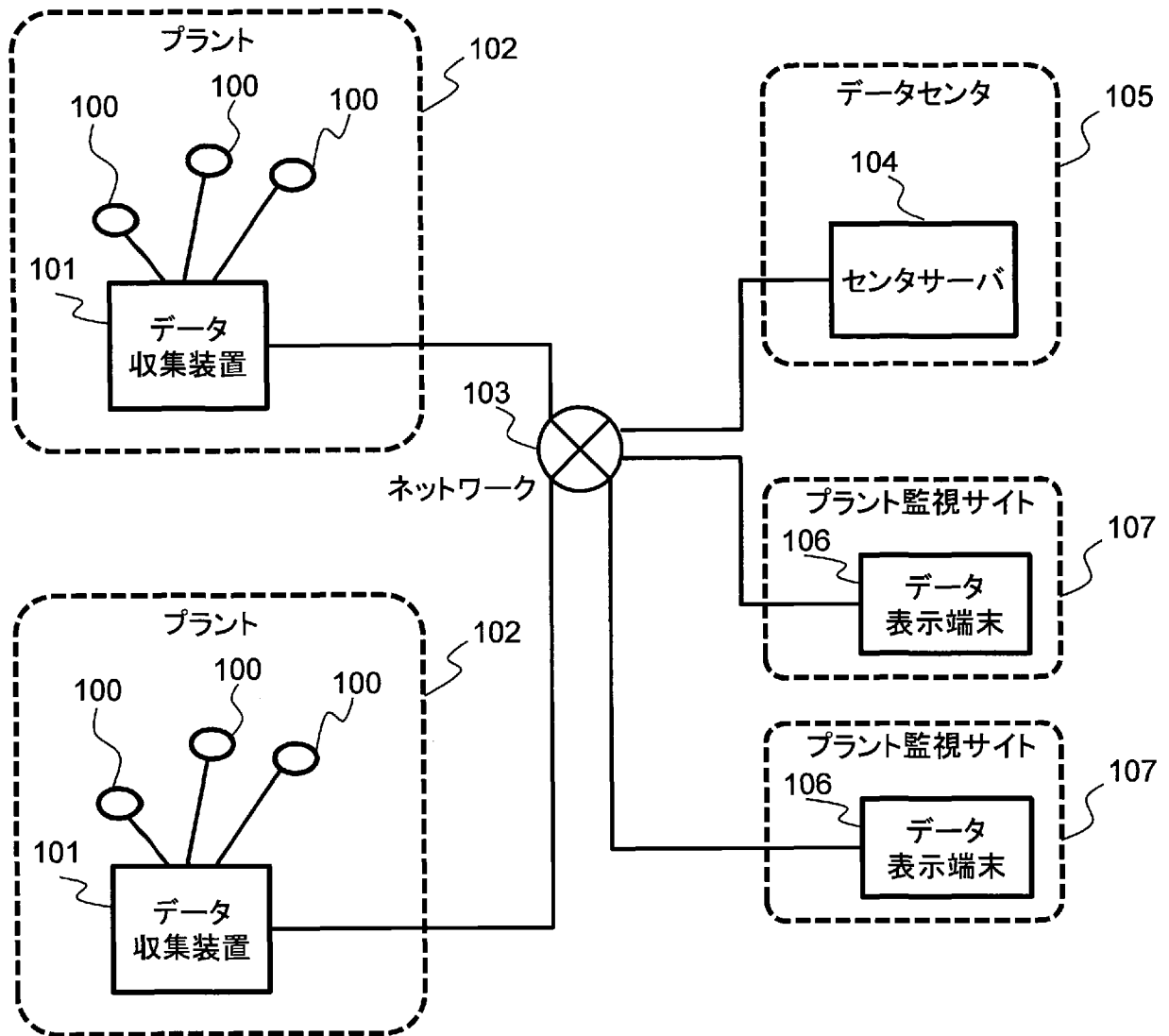
利用者の最も利用する検索期間を統計処理により生成することにより前記時系列ブロックの分割範囲を得る請求項 21 に記載の時系列データ管理方法。

[請求項23]

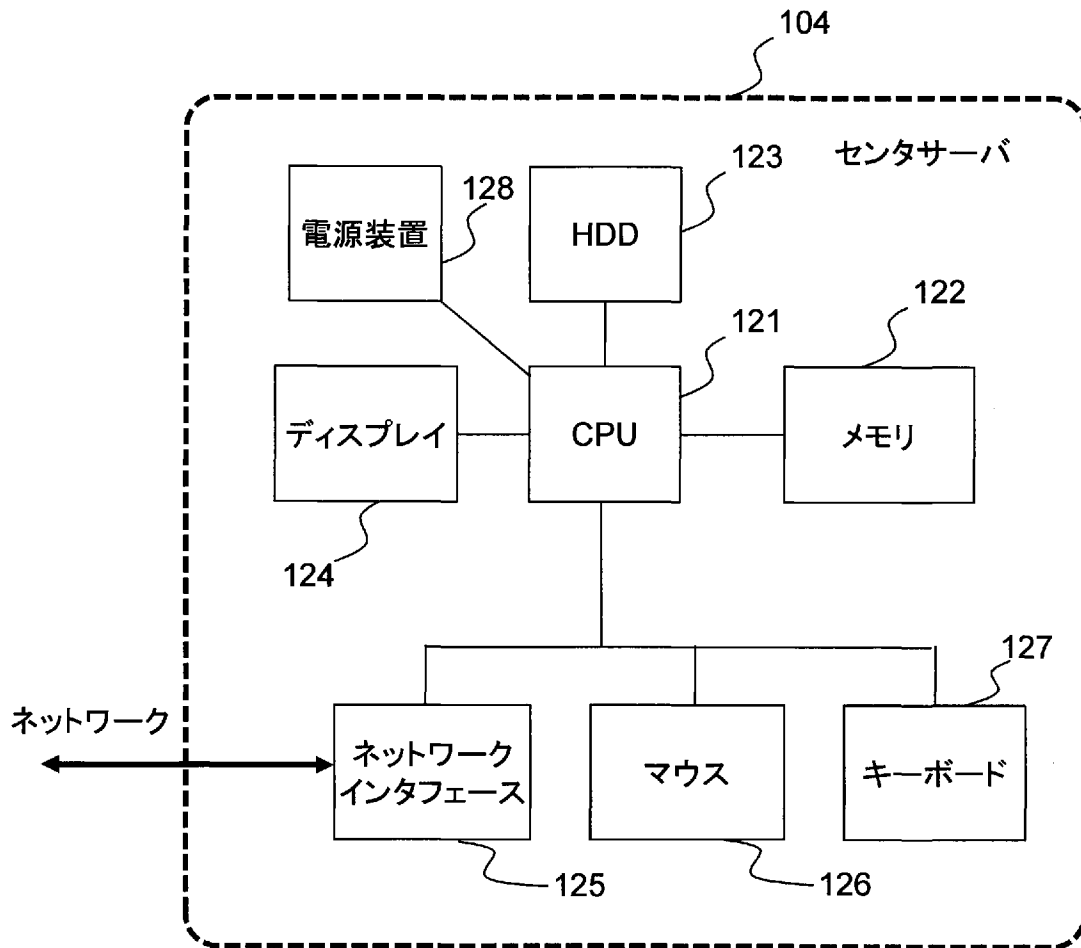
時系列の特徴変化が起こる最小期間を統計処理により生成することにより前記時系列ブロックの分割範囲を得る請求項 21 に記載の時系列データ管理方法。

- [請求項24] 前記時系列データ格納手段では時系列ブロックを圧縮する手段を、前記第二の検索手段では圧縮された時系列ブロックを展開する手段を備えることを特徴とする請求項21に記載の時系列データ管理方法。
- [請求項25] 前記ユーザからの問い合わせが閾値超過ないし未滿判定の場合、前記第一の検索手段で最大値ないし最小値との超過判定ないし未滿判定で時系列ブロックを絞り込むことを特徴とする請求項21に記載の時系列データ管理方法。
- [請求項26] 時系列データに対して時系列処理を行うことで特徴量を生成し、時系列を時系列ブロックに分割し、該特徴量を検索キーとなるメタ情報として時系列ブロックとともに時系列蓄積装置に格納する時系列データ格納手段と、
ユーザからの問い合わせに応じて該特徴量を検索キーに時系列ブロックを検索する第一の問合せと、時系列ブロック内の時系列を検索する第二の問い合わせを生成する手段と、
第一の問い合わせにより該時系列ブロックを検索する第一の検索手段と、
得られた時系列ブロックを逐次第二の問い合わせに基づき時系列処理を行って時系列を検索し、問合せ結果を出力する第二の検索手段と、
を有することを特徴とする計算機。
- [請求項27] リアルタイムに到来する時系列データを所定期間メモリ上に蓄積し、時系列解析を行うことで前記特徴量を生成するストリームデータ処理エンジンを備えることを特徴とする請求項26に記載の計算機。
- [請求項28] 前記第二の検索手段において、移動平均、区間抽出、間引き処理の少なくともいずれかを行うストリームデータ処理エンジンを備えることを特徴とする請求項26に記載の計算機。

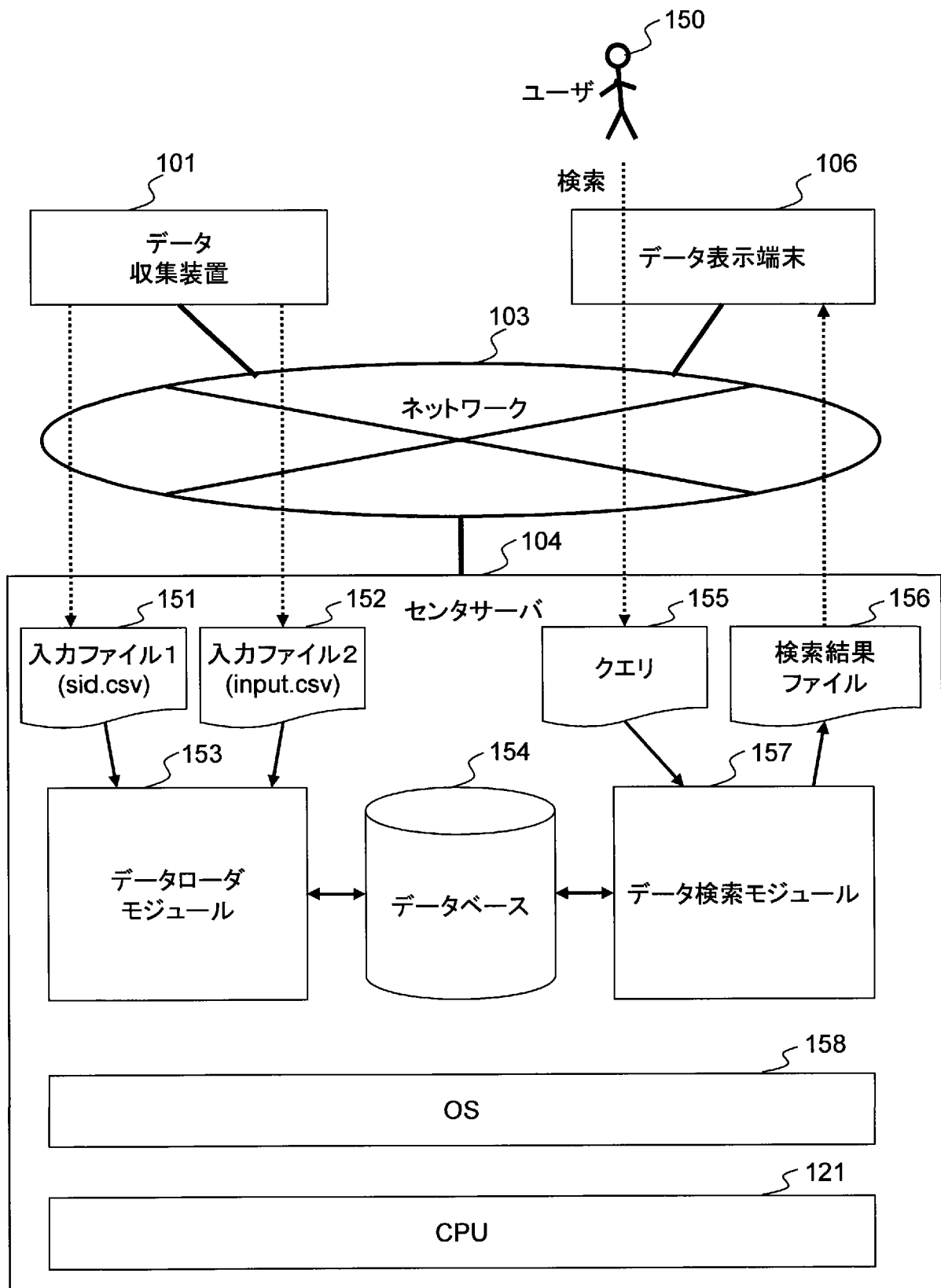
[図1]



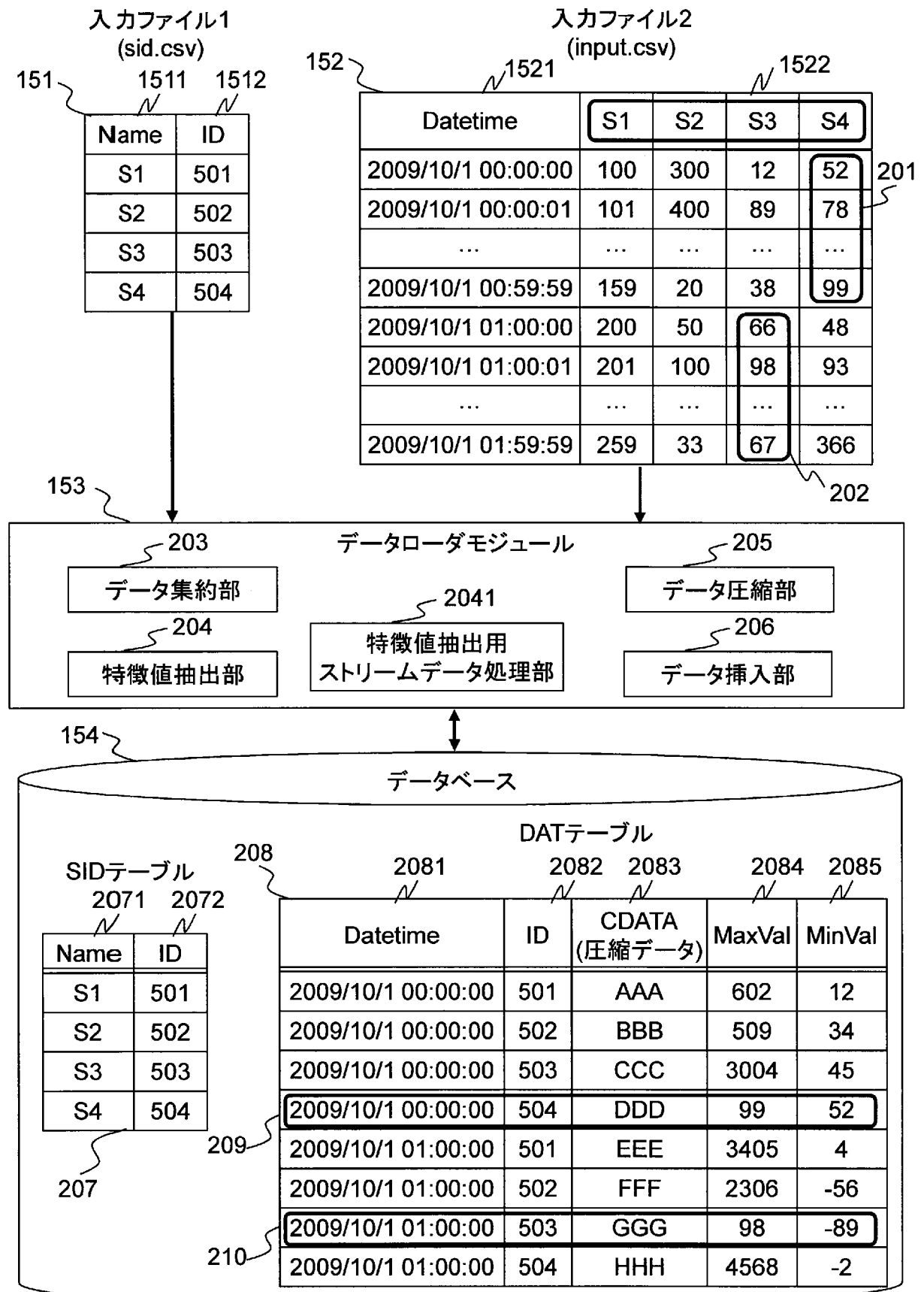
[図2]



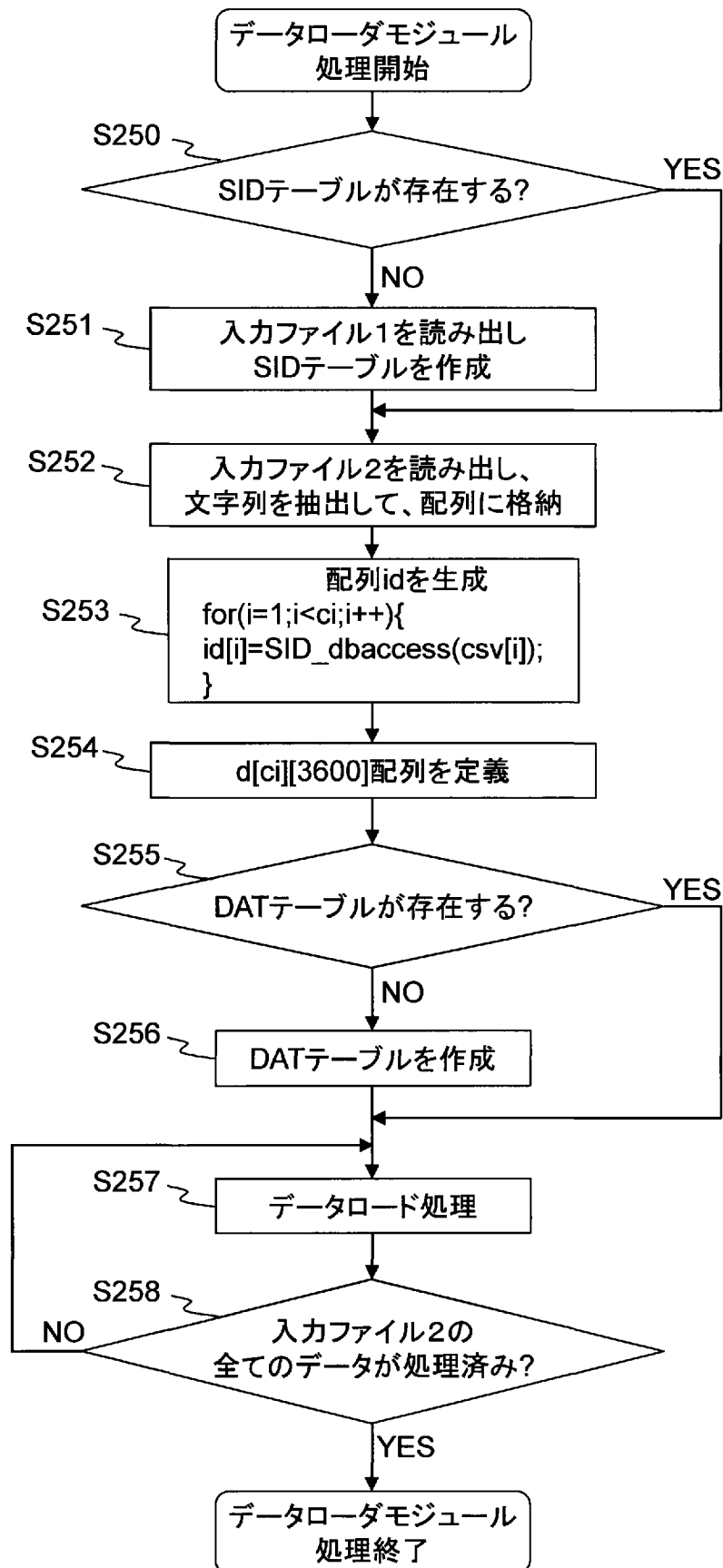
[図3]



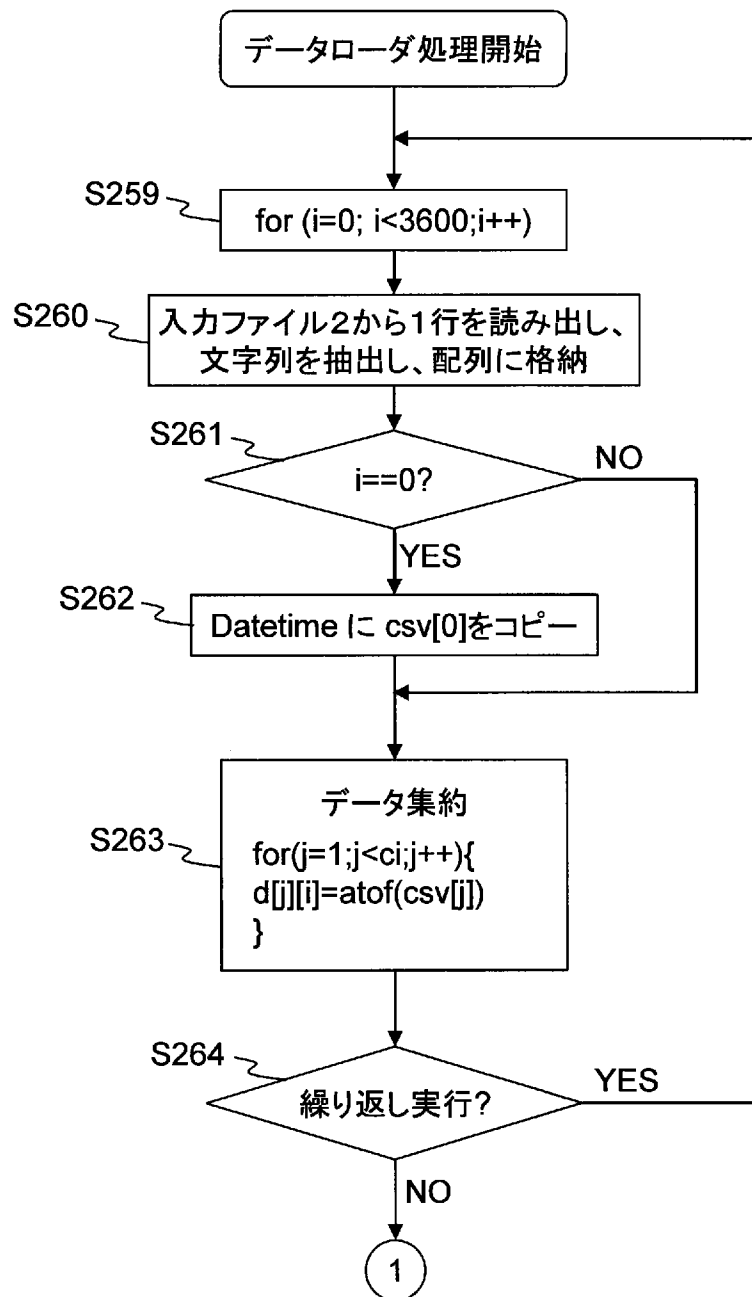
[図4]



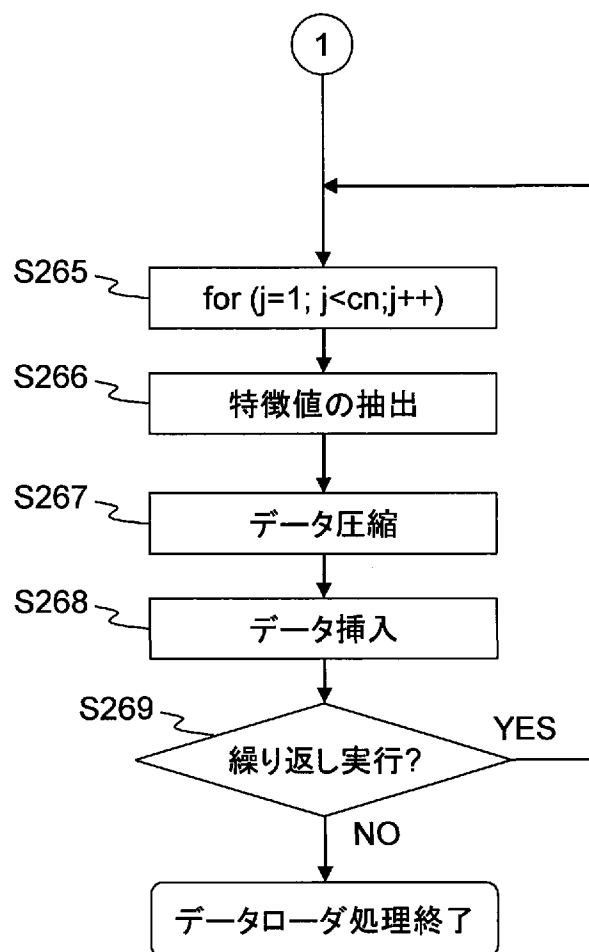
[図5A]



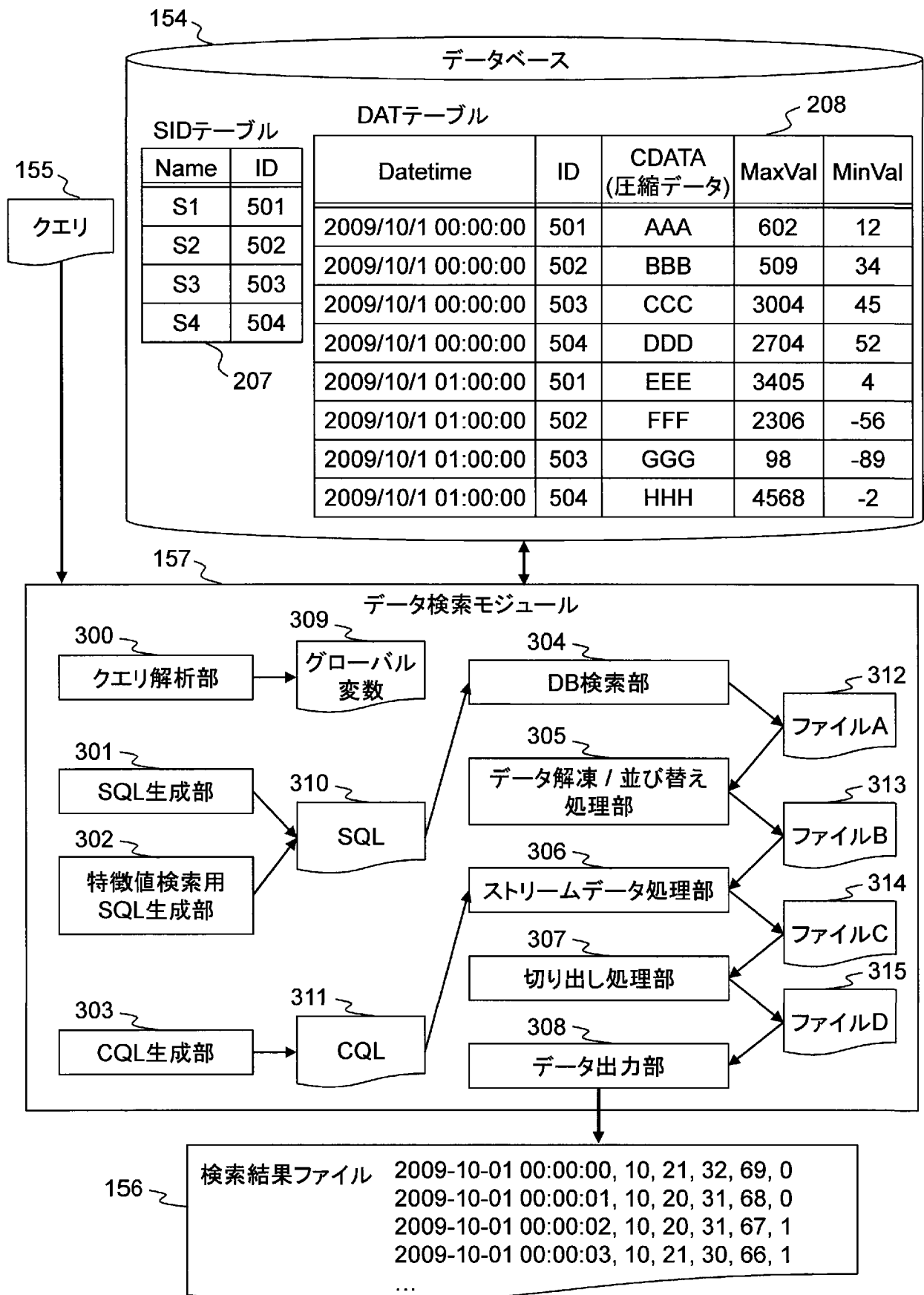
[図5B]



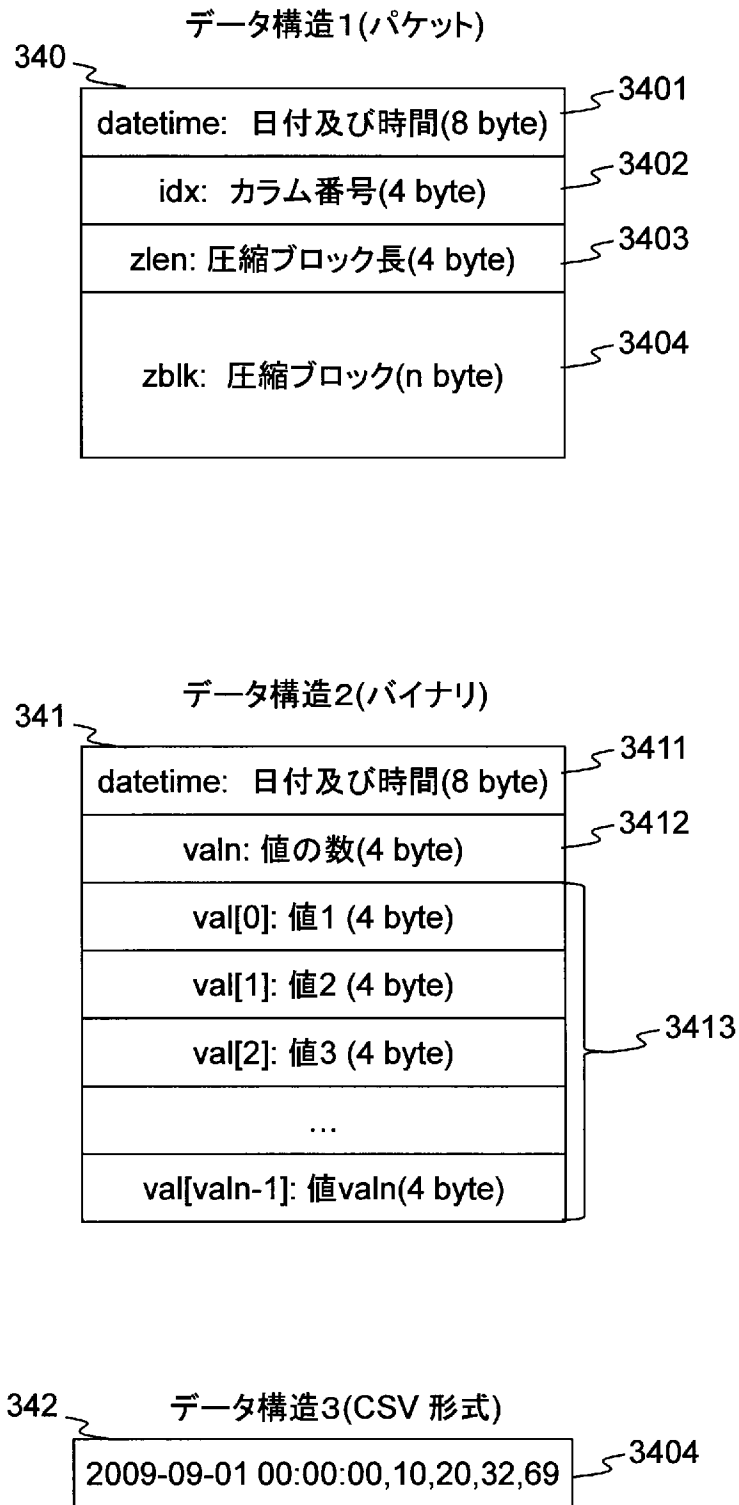
[図5C]



[図6]



[図7]



[図8A]

クエリ例1

350

```
input_items: 'Sensor1', 'Sensor2' ;
select_range: 5;
select_items:
    'Sensor1' as L1 ,
    'Sensor2' as L2 ,
    avg('Sensor2') as L3 ;
where_timerange: 2009-02-02 00:00:00, 2009-02-20 23:59:59 ;
where_condition:
    ortest(
        GreaterThanOrEqualTo(L1, 70),
        PositiveThreshold(L3, 300, 200)
    );
output_range: 60,60;
step: 5 ;
file: result.csv ;
```

クエリ例2

351

```
input_items: 'Sensor1', 'Sensor2', 'Sensor3' ;
where_timerange: 2009-02-02 00:00:00, 2009-02-20 23:59:59 ;
file: result.csv ;
```

[図8B]

SQL例1

352

```

select Datetime, SID.Name, DAT.CDATA
from SID join DAT on SID.ID = DAT.ID
where Name in ( 'Sensor1', 'Sensor2' ) and
      Datetime between '2009-02-01 23:00:00' and
'2009-02-20 23:59:59'
order by Datetime;

```

CQL例1

353

```

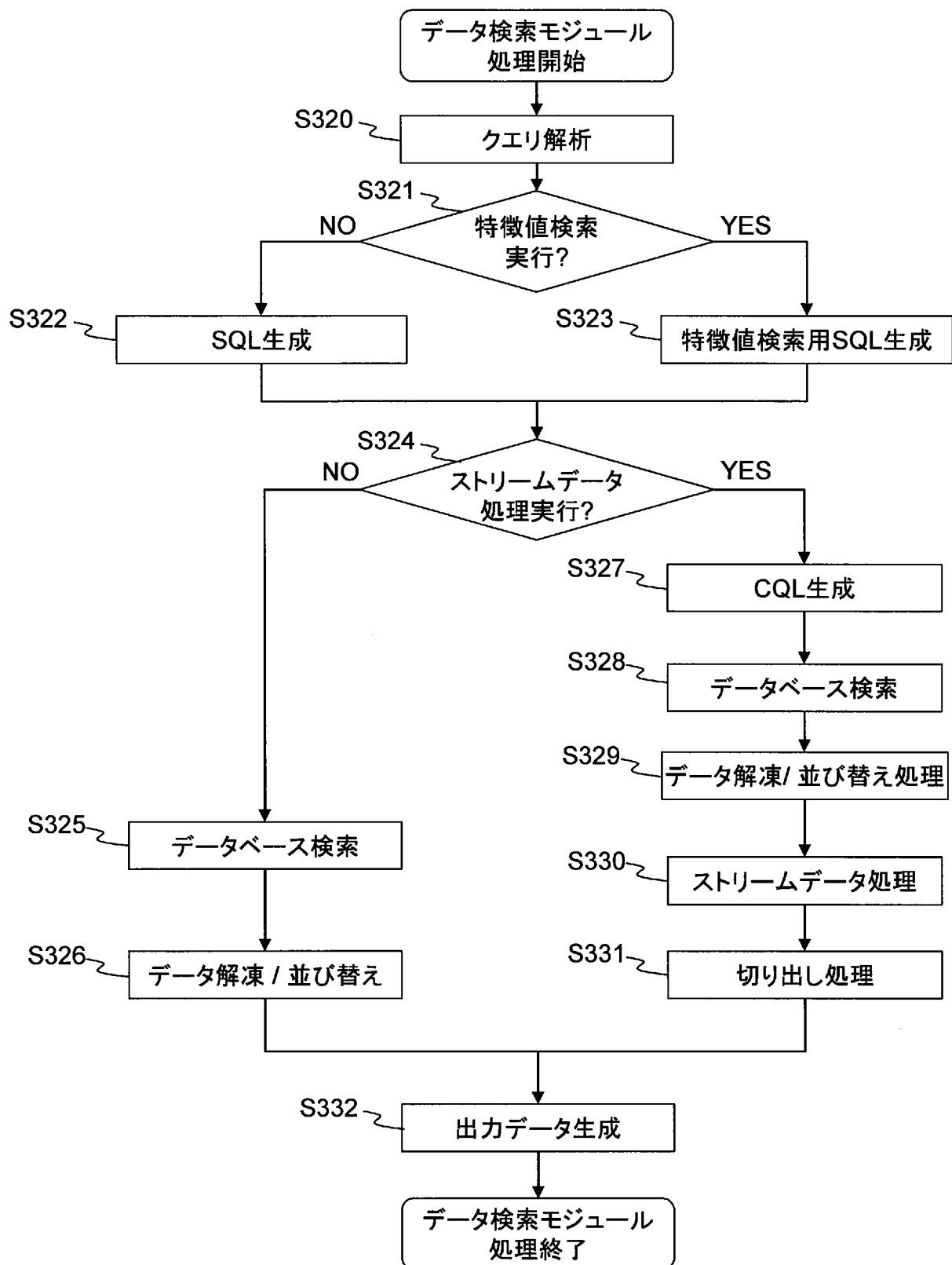
register stream IN (
  ts__ TIMESTAMP
  , 'Sensor1' REAL, 'Sensor2' REAL);

register query GFUNG
  select last(ts__) as L0__
  , last('Sensor1') as L1 , last('Sensor2') as L2 , avg('Sensor2')
as L3
from IN [range 5 second ];

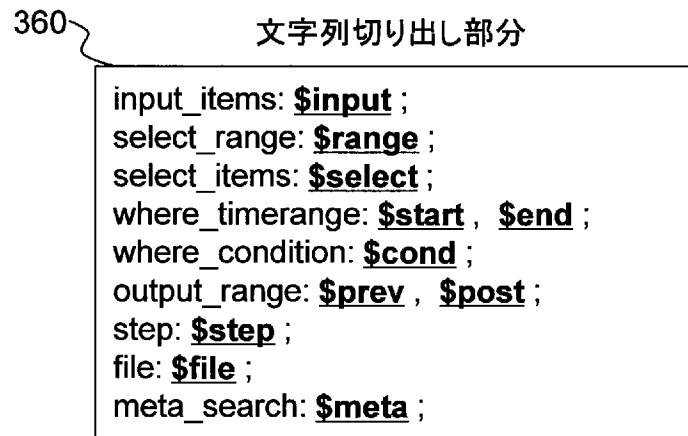
register query OUT istream(
  select L0__
  , L1, L2, L3
  ,
  ortest( GreaterThanOrEqualTo(L1, 70), PositiveThreshold(L3,
300, 200)
  ) as CONDITION
from GFUNG);

```

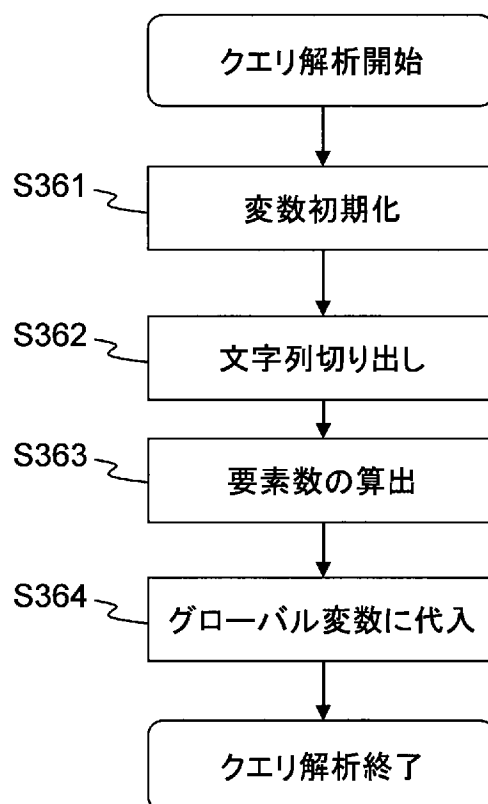
[図9]



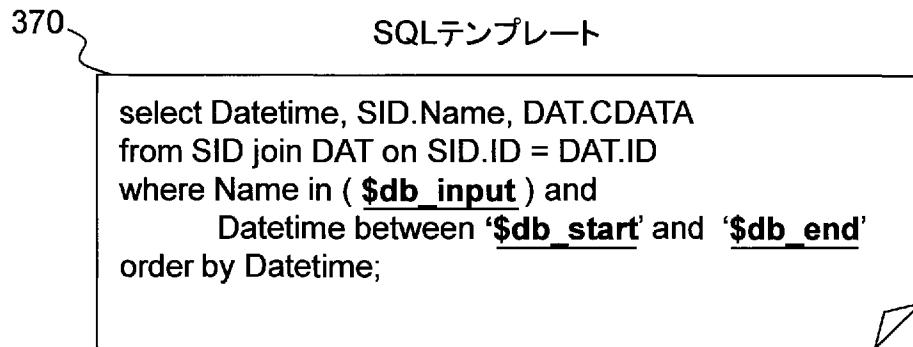
[図10A]



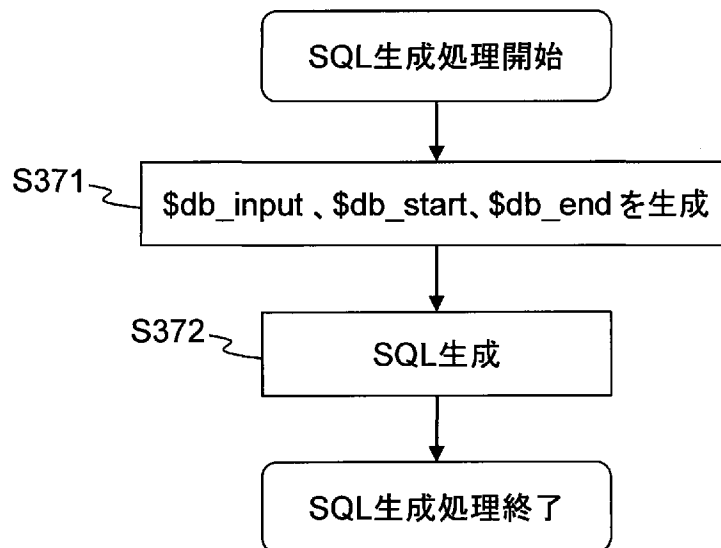
[図10B]



[図11A]



[図11B]



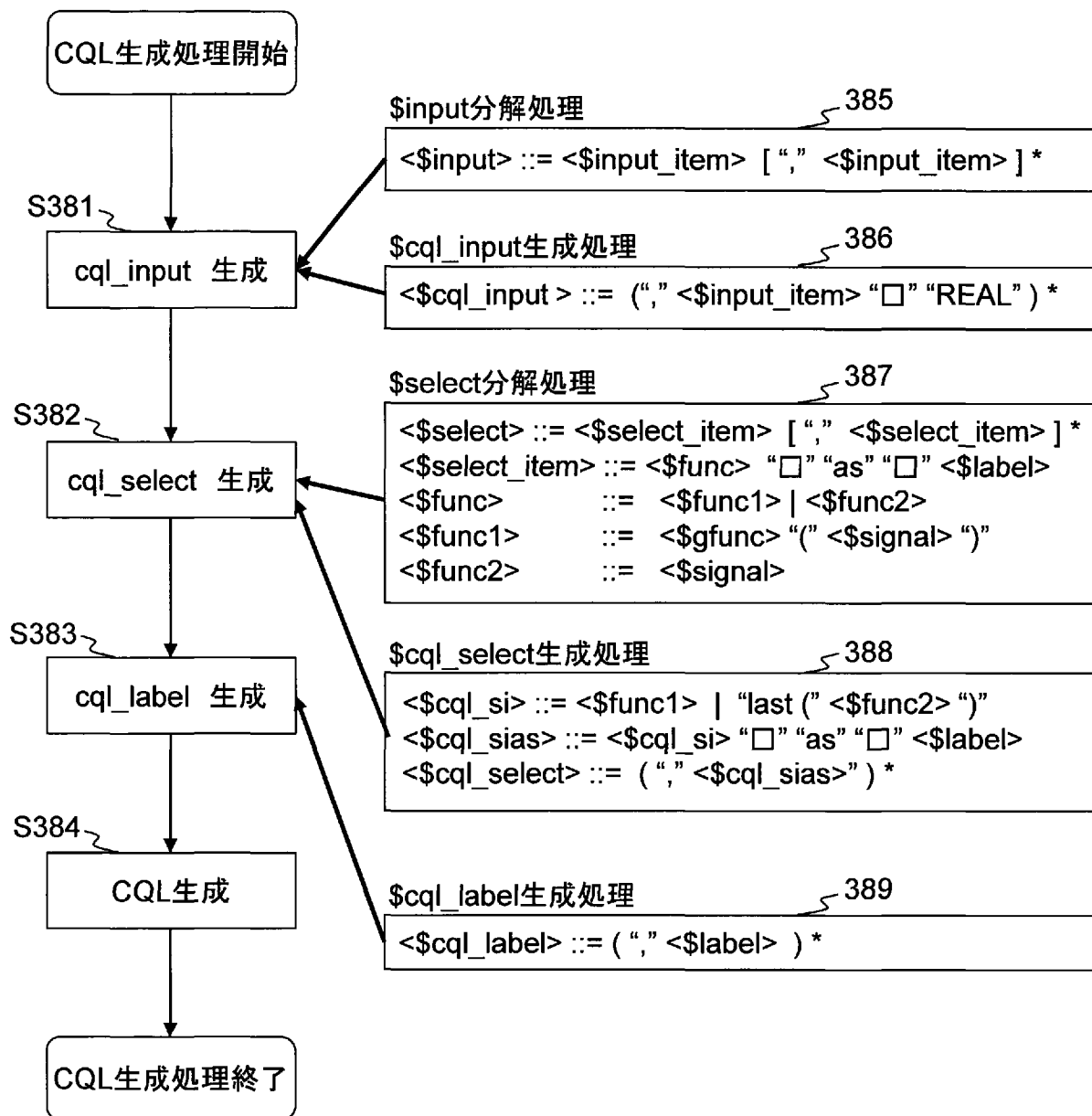
[図12A]

380

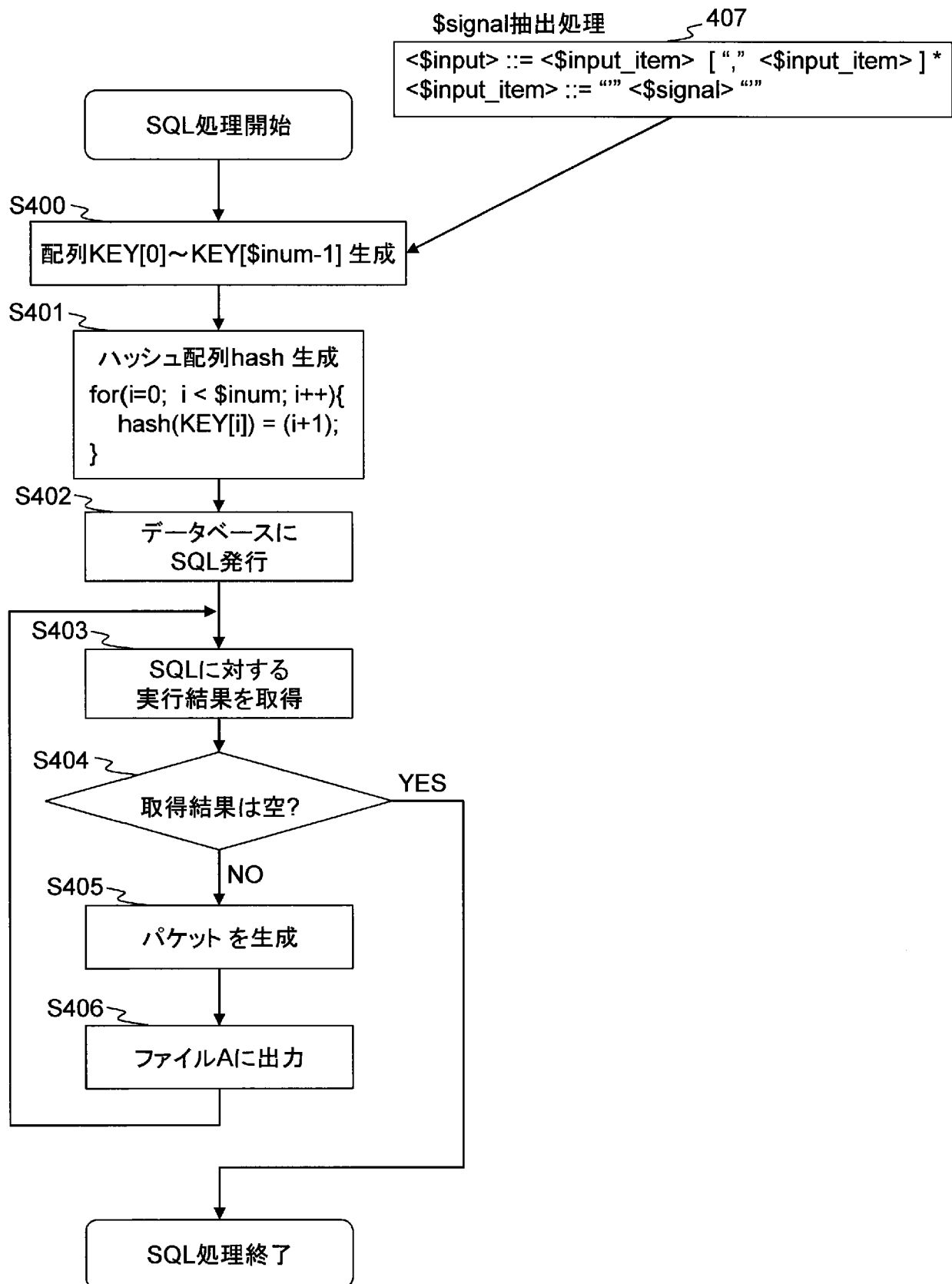
CQLテンプレート

```
register stream IN (  
  ts__TIMESTAMP  
  $cql_input);  
  
register query GFUNG  
  select last(ts__) as L0__  
  $cql_select  
from IN [range $range second ];  
  
register query OUT istream(  
  select L0__  
  $cql_label  
  ,  
  $cond as CONDITION  
from GFUNG);
```

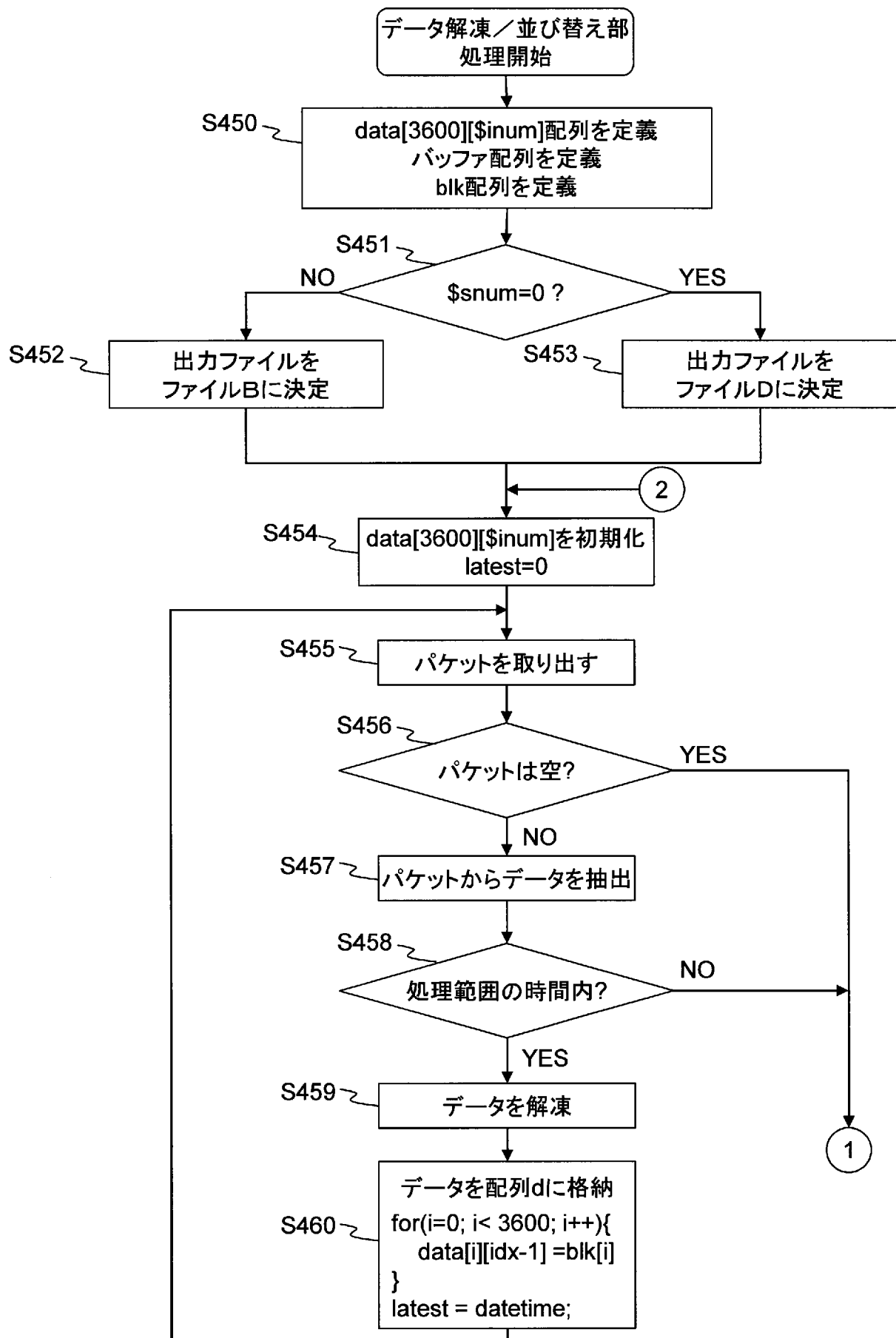
[図12B]



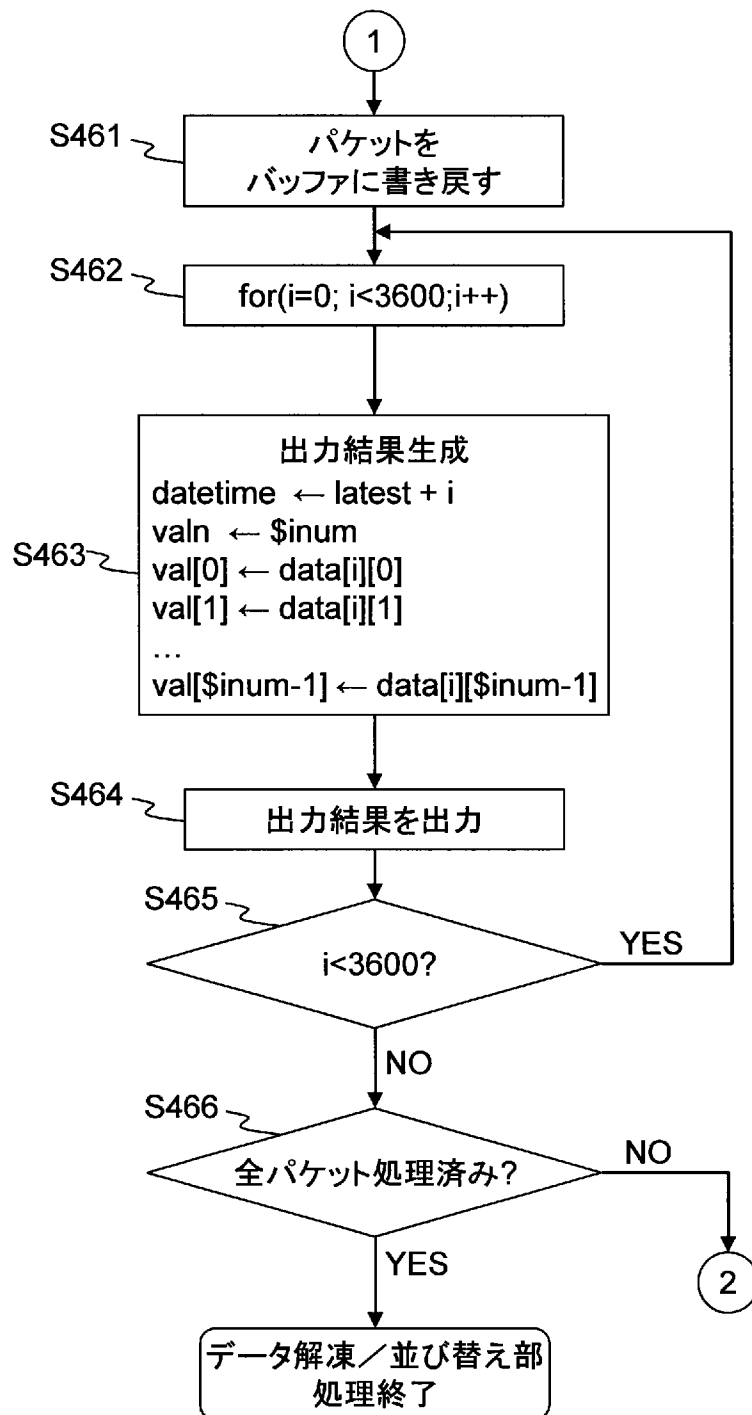
[図13]



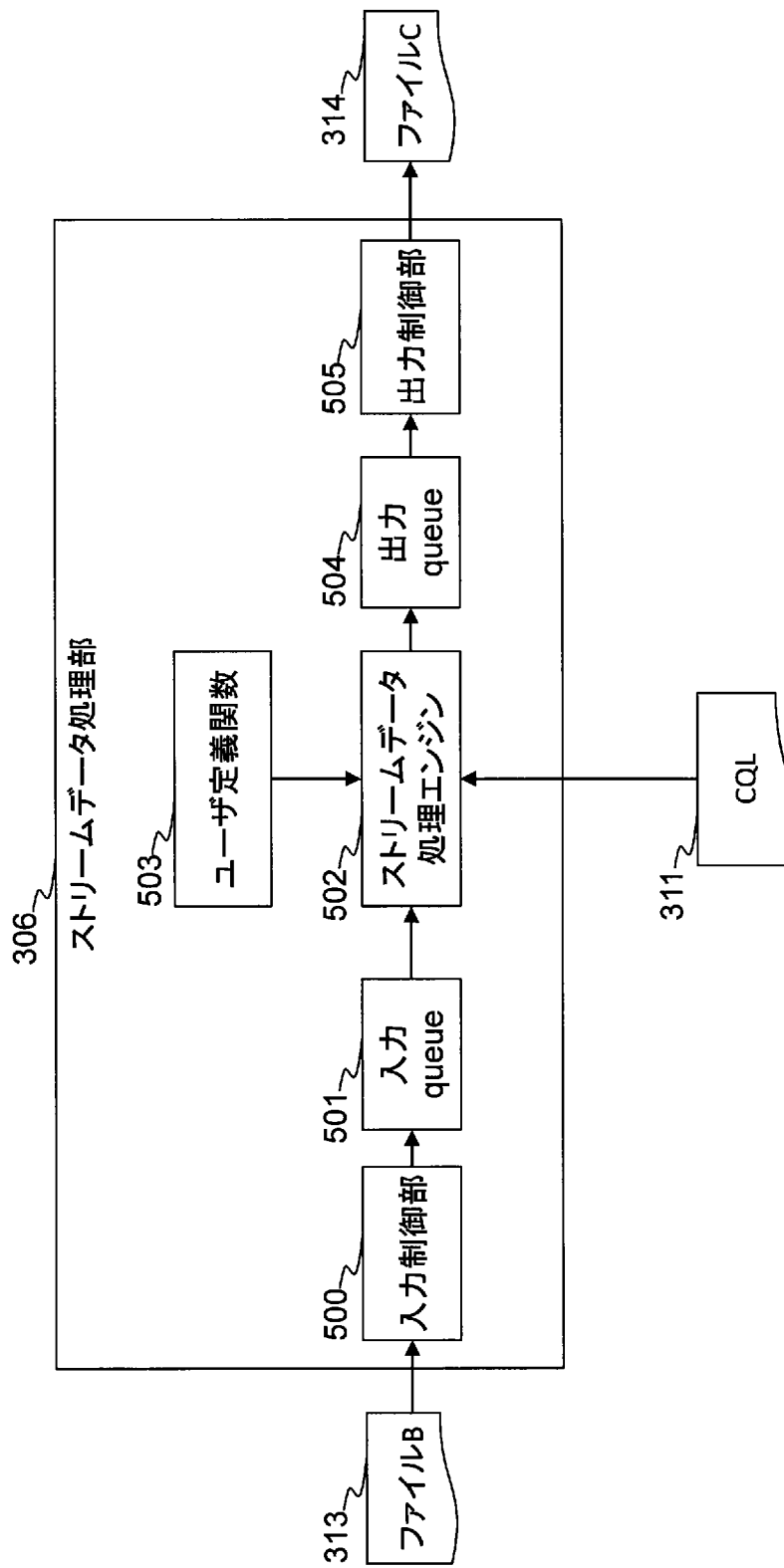
[図14A]



[図14B]



[図15]



[図16]

503 ~

5031

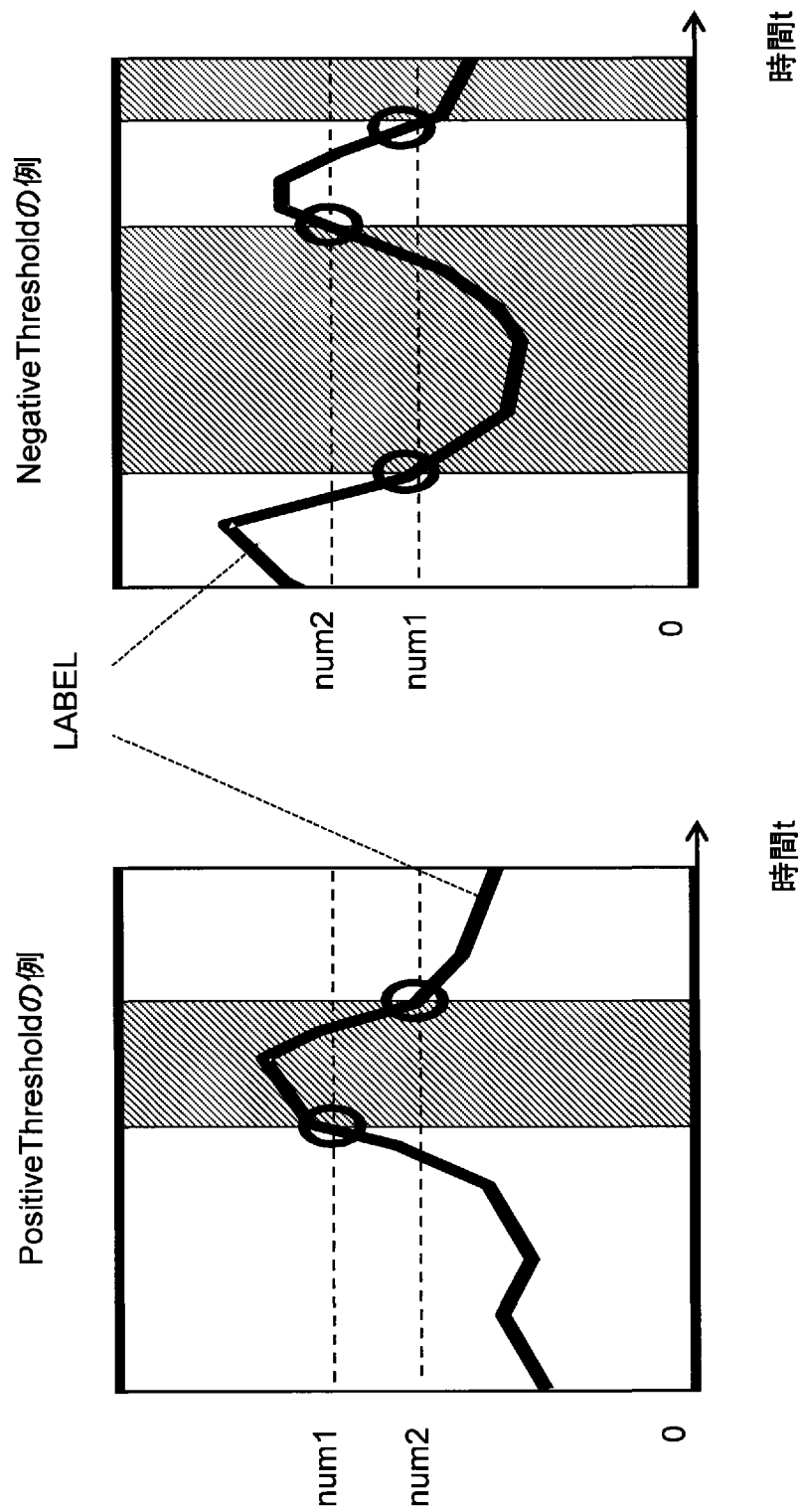
5032

5033

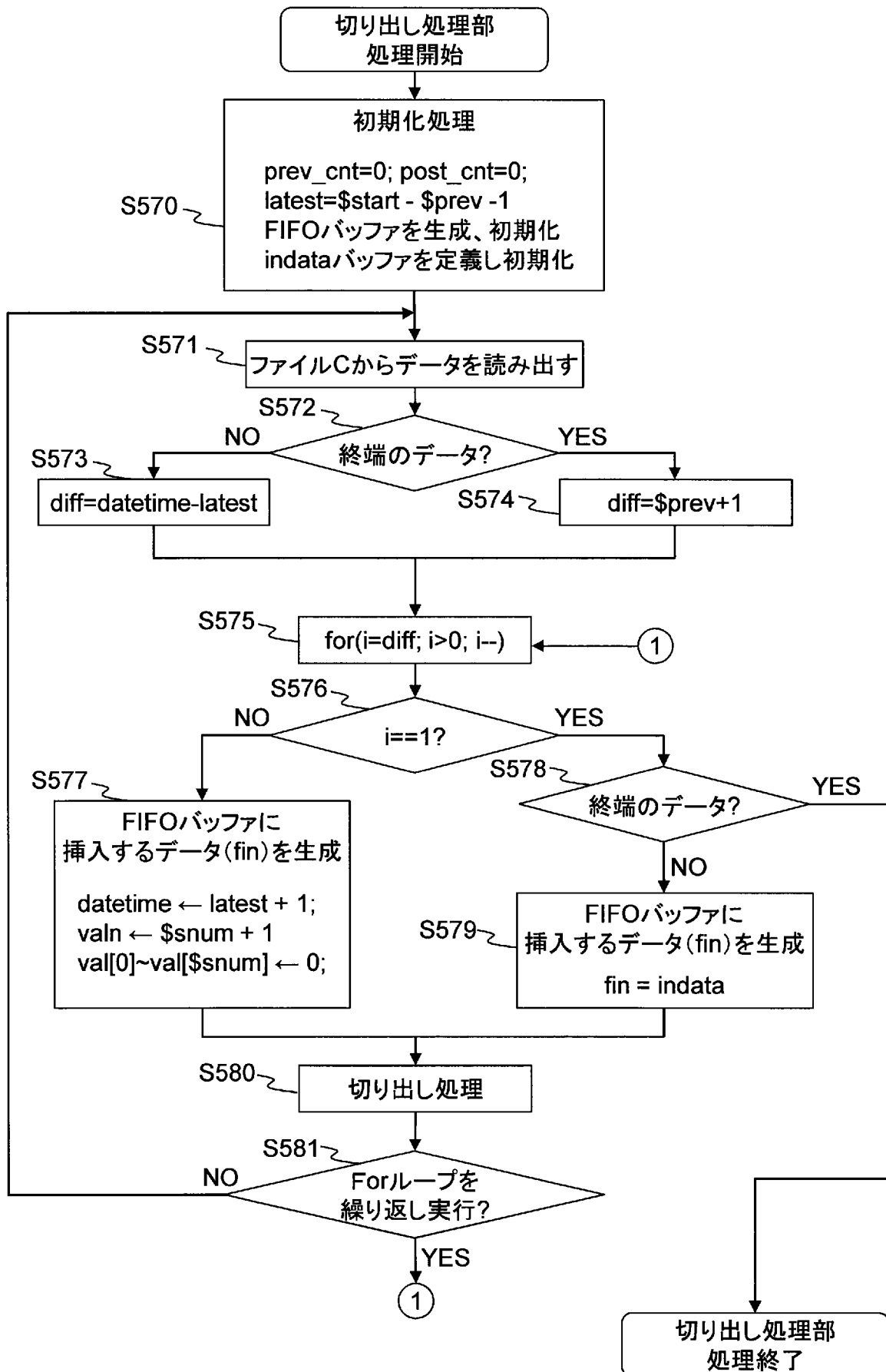
5034

#	関数名	動作説明	備考
1	last(LABEL)	ウィンドウ内の最終の値を返す関数	集約関数
2	Graterthan(LABEL, num)	(LABEL > num)	
3	GraterthanOrEqualTo(LABEL, num)	(LABEL >= num)	
4	Lessthan (LABEL, num)	(LABEL < num)	
5	LessthanOrEqualTo(LABEL, val)	(LABEL <= num)	
6	EqualTo(LABEL, val)	(LABEL == num)	
7	NotEqualTo(LABEL, val)	(LABEL != num)	
8	PositiveThreshold (LABEL, num1, num2)	LABEL>=num1からLABEL<num2までの 区間はtrue	
9	NegativeThreshold (LABEL, num1, num2)	LABEL<=num1からLABEL>num2までの区 間はtrue	
10	andTest(LABEL1, LABEL2)	(LABEL1 && LABEL2)	
11	orTest(LABEL1, LABEL2)	(LABEL1 LABEL2)	
12	notTest(LABEL)	(!LABEL)	

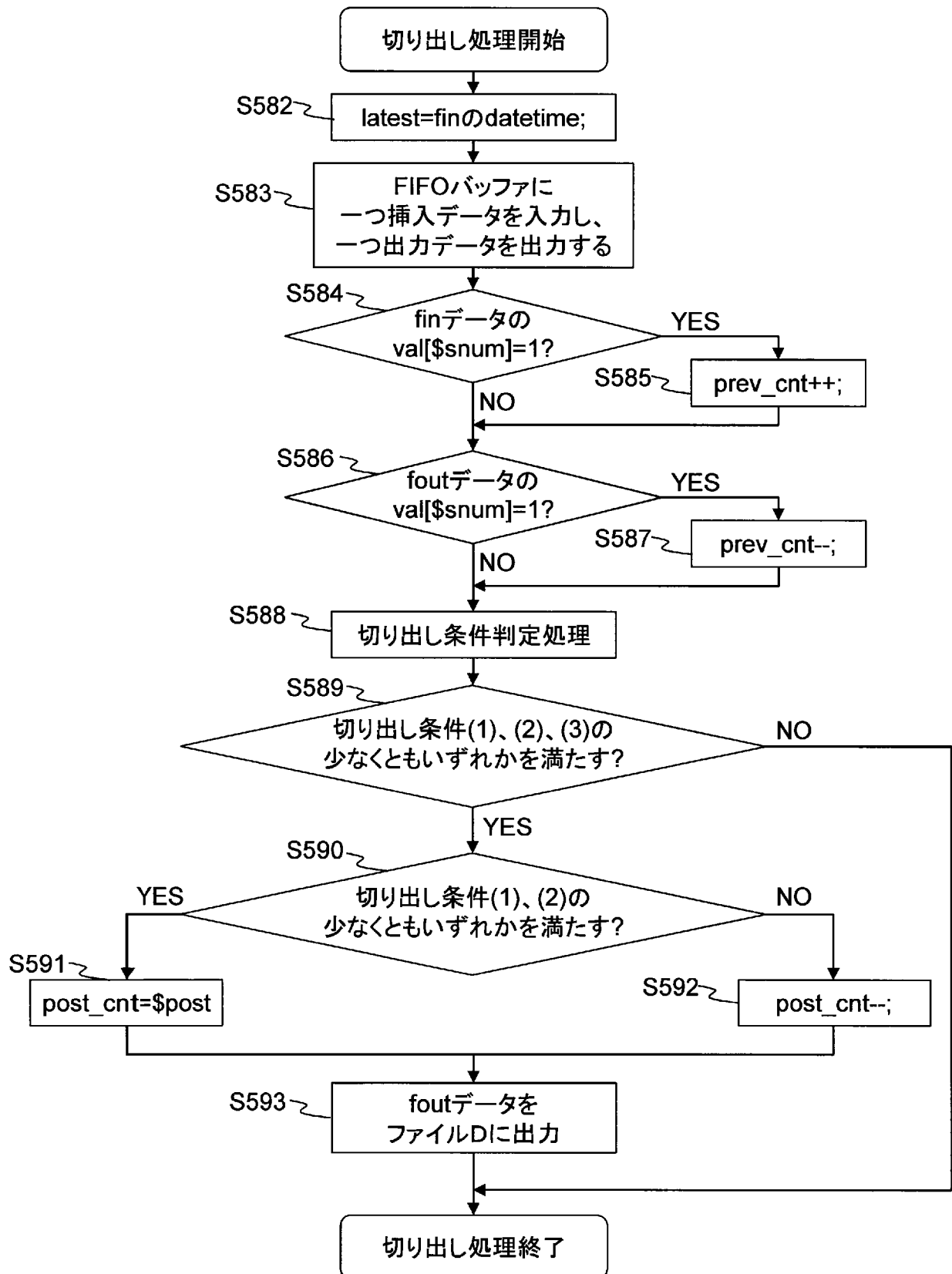
[図17]



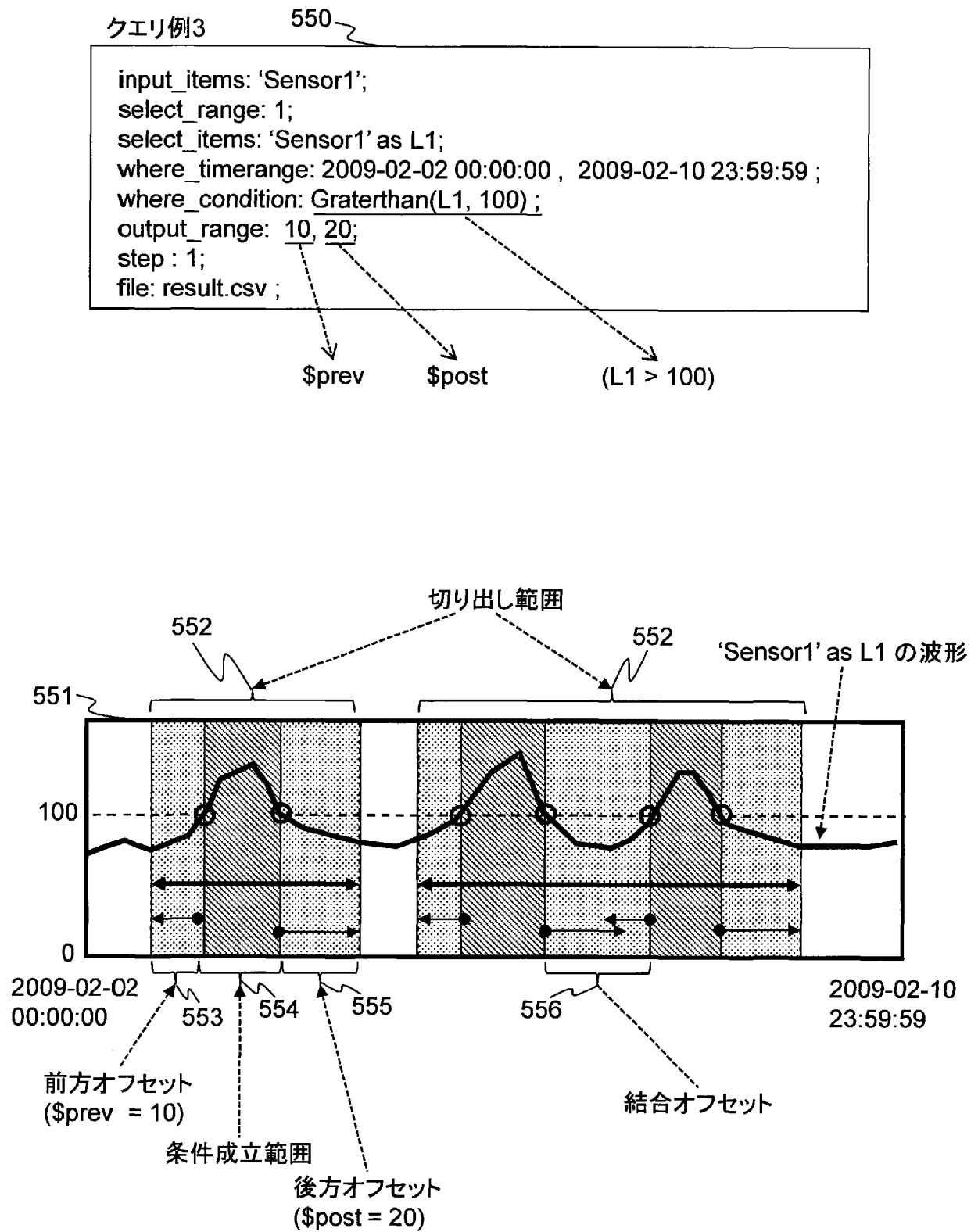
[図18A]



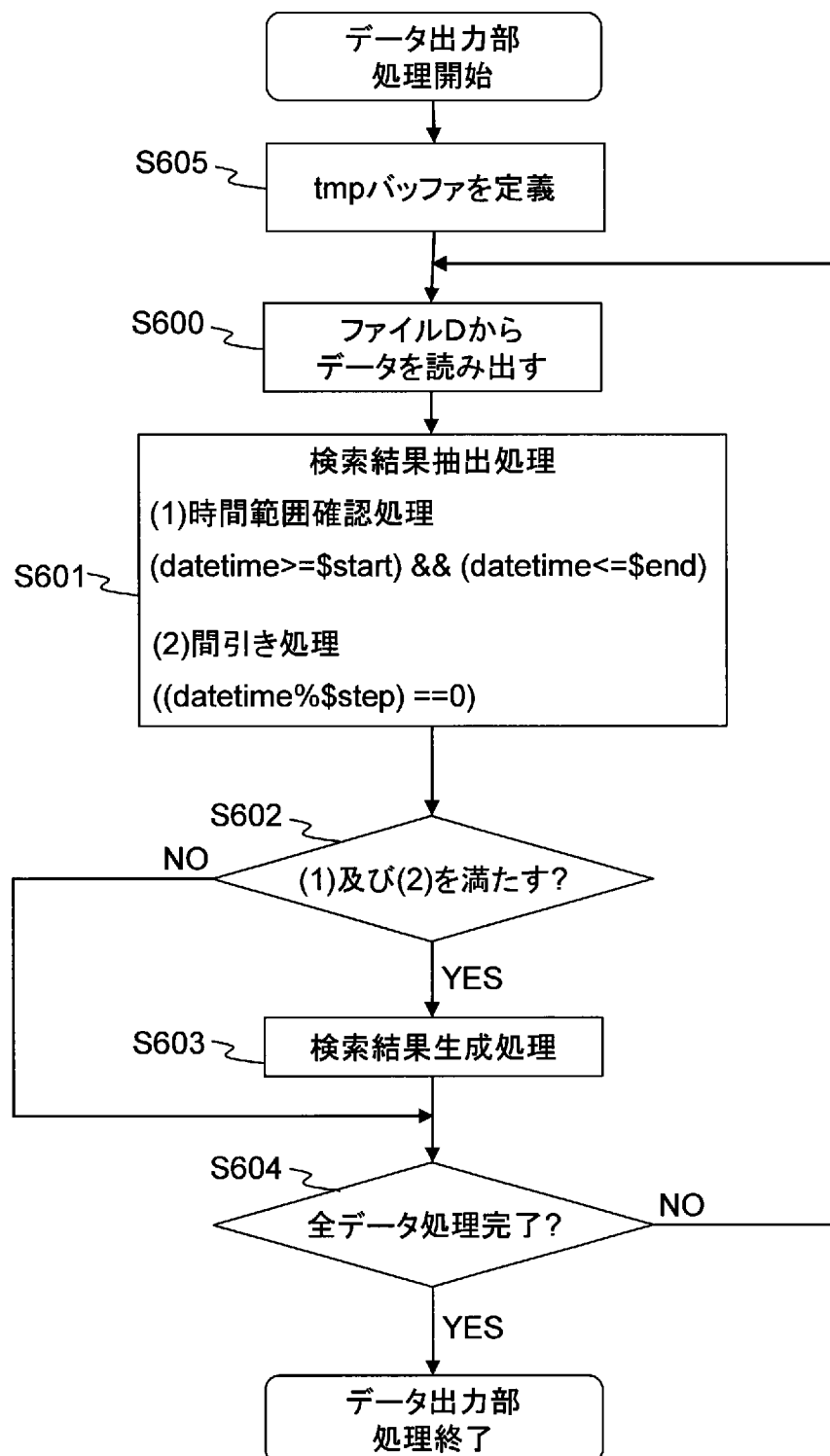
[図18B]



[図19]



[図20]



[図21]

クエリ例4

700

```

input_items: 'Sensor1', 'Sensor2', 'Sensor3', 'Sensor4', 'Sensor5', 'Sensor6' ;
select_range: 5;
select_items:
    'Sensor1' as L1 ,
    'Sensor2' as L2 ,
    avg('Sensor2') as L3 ;
    'Sensor3' as L4,
    'Sensor4' as L5 ,
    'Sensor5' as L6 ,
    'Sensor6' as L7 ;

where_timerange: 2009-01-01 00:00:00, 2009-12-31 23:59:59 ;
where_condition: graterthan(L1, 100) ;

output_range:60,60;
step: 5 ;
file: result.csv ;
meta_search: 'Sensor1'.MaxVal > 100 ;

```

SQL例4 - 1

701

```

select Datetime
from SID join DAT on SID.ID = DAT.ID
where (Name = 'Sensor1' and MaxVal > 100 ) and
      Datetime between '2009-01-01 00:00:00' and '2009-12-25 23:59:59'
order by Datetime;

```

SQL例4 - 2

702

```

select Datetime, SID.Name, DAT.CDATA
from SID join DAT on SID.ID = DAT.ID
where Name in ( 'Sensor1', 'Sensor2' ) and
      ( ( Datetime between '2009-02-04 11:00:00' and '2009-02-04 15:59:59' ) or
        ( Datetime between '2009-04-11 02:00:00' and '2009-04-11 05:59:59' ) or
        ( Datetime between '2009-07-18 15:00:00' and '2009-07-18 19:59:59' ) )
order by Datetime;

```

[図22]

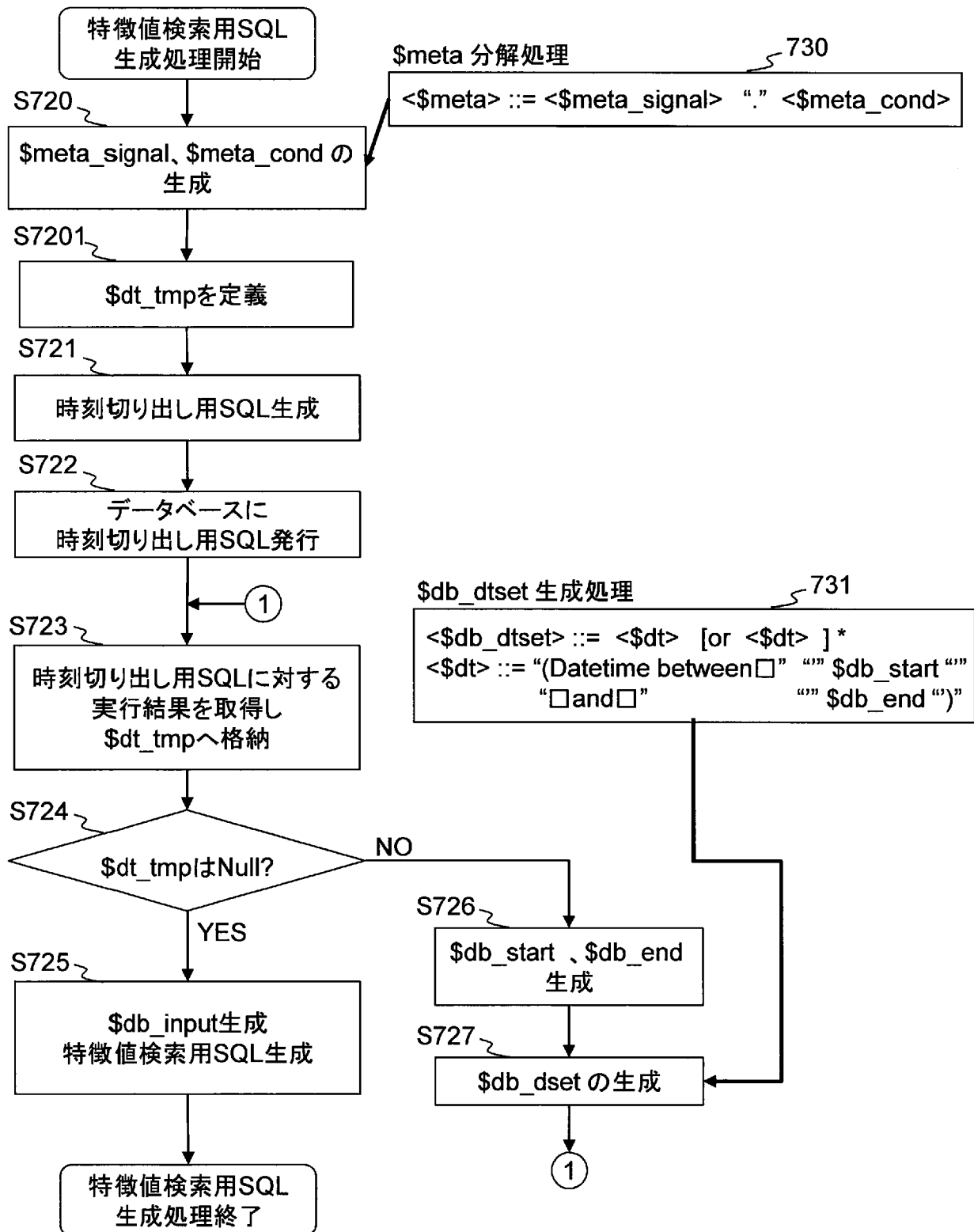
710 時刻切り出し用SQLテンプレート

```
select Datetime
from SID join DAT on SID.ID = DAT.ID
where (Name = $meta_signal and $meta_cond)
      and Datetime
      between '$meta_start' and '$meta_end'
order by Datetime;
```

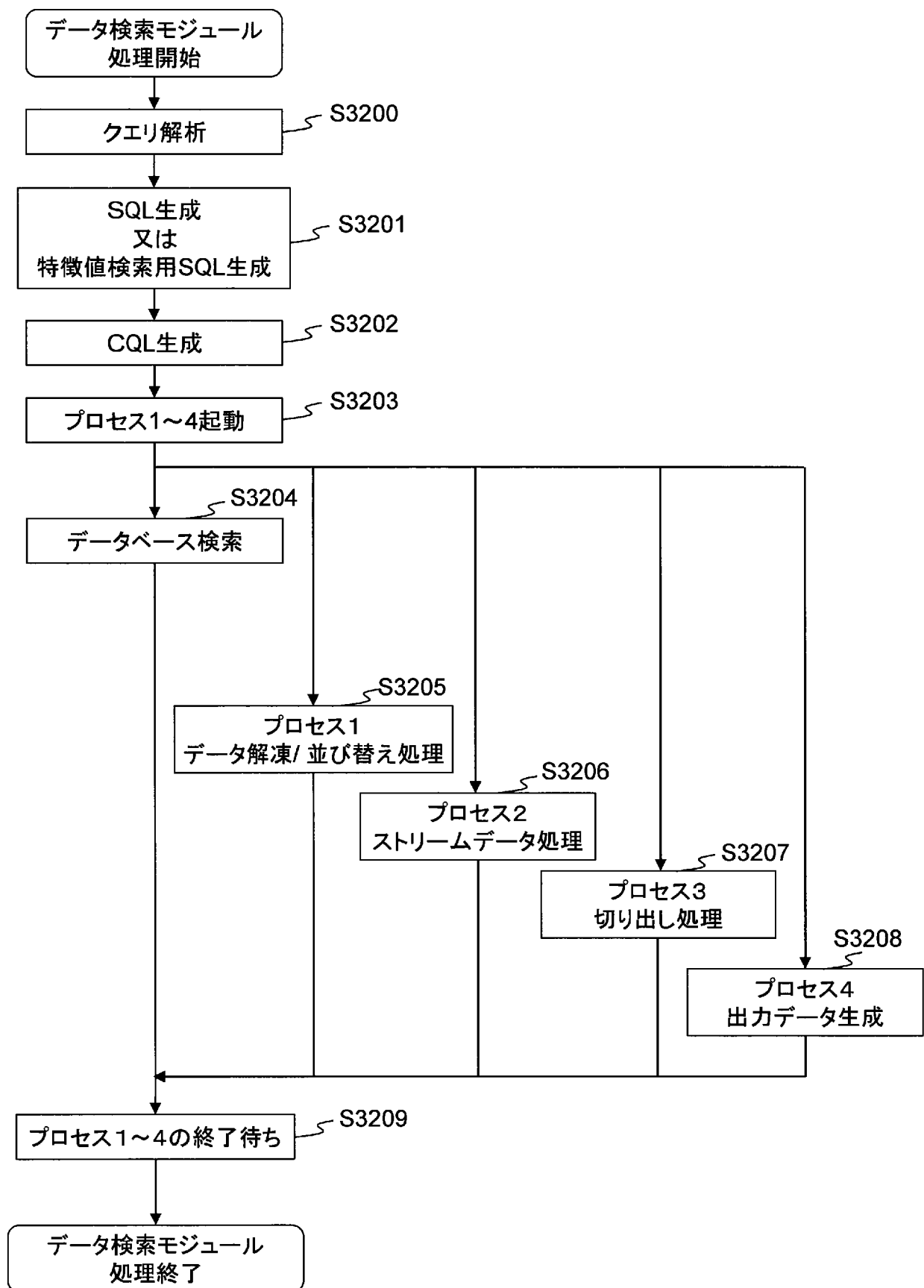
711 特徴値検索用SQLテンプレート

```
select Datetime, SID.Name, DAT.CDATA
from SID join DAT on SID.ID = DAT.ID
where Name in ( $db_input ) and
              ($db_dtset)
order by Datetime;
```

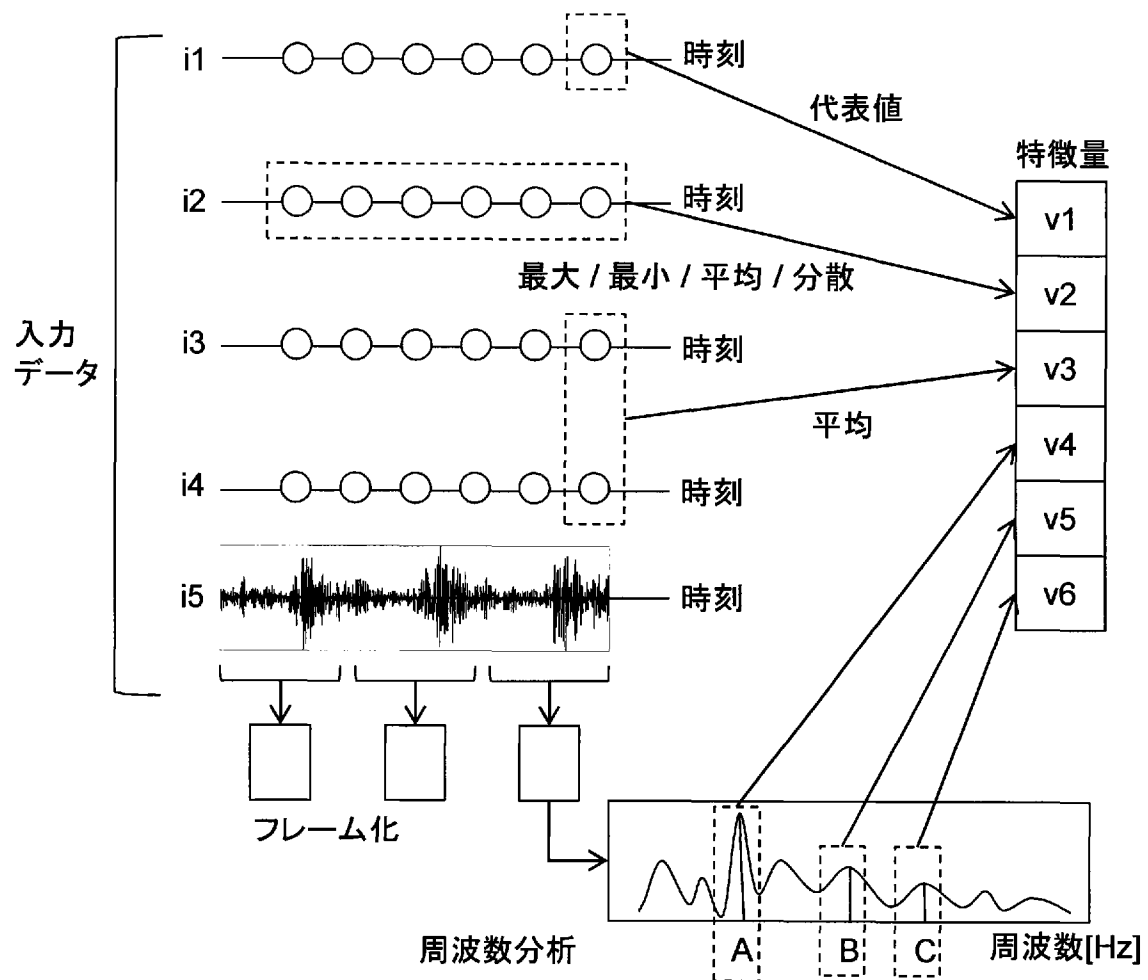
[図23]



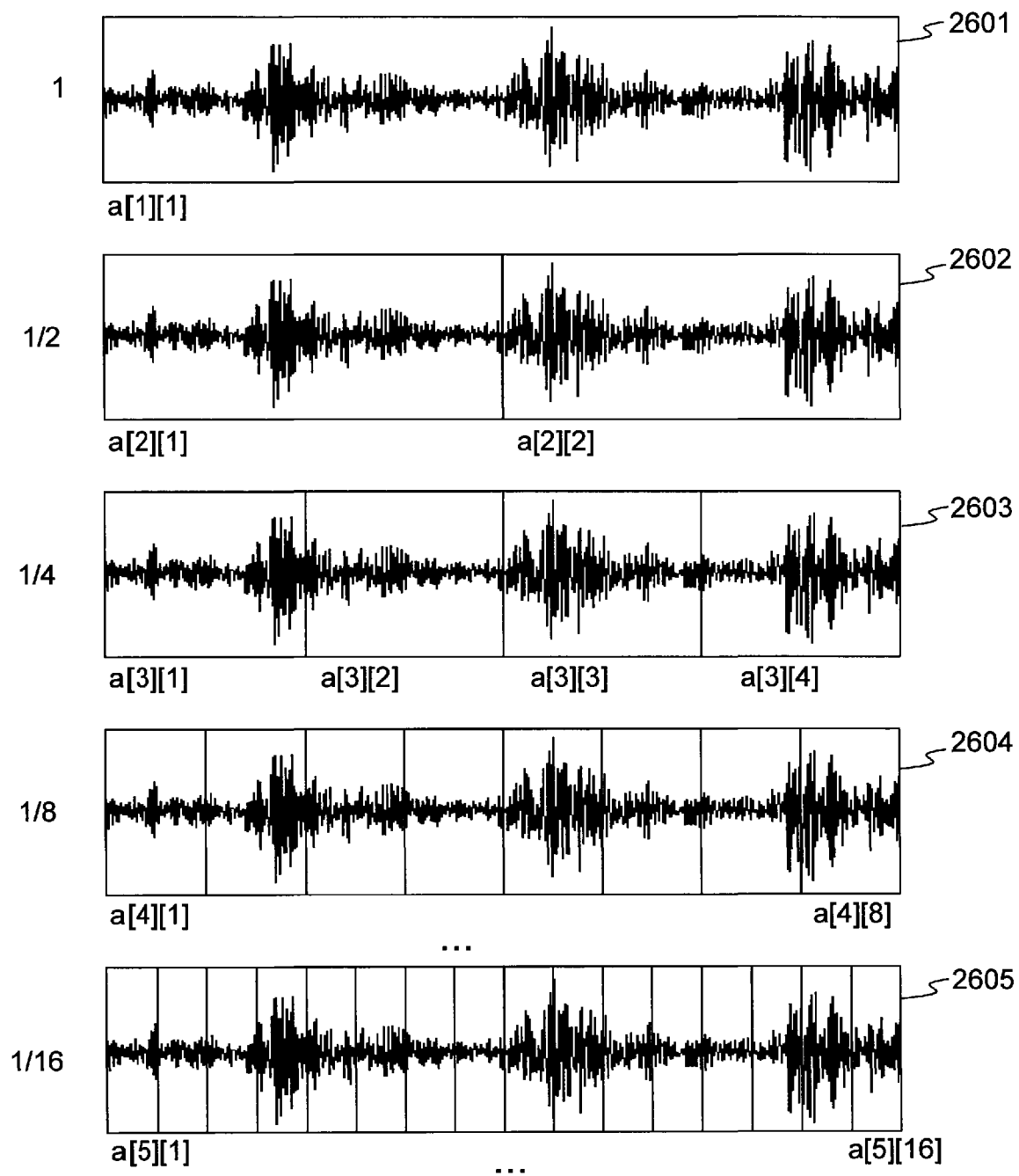
[図24]



[図25]

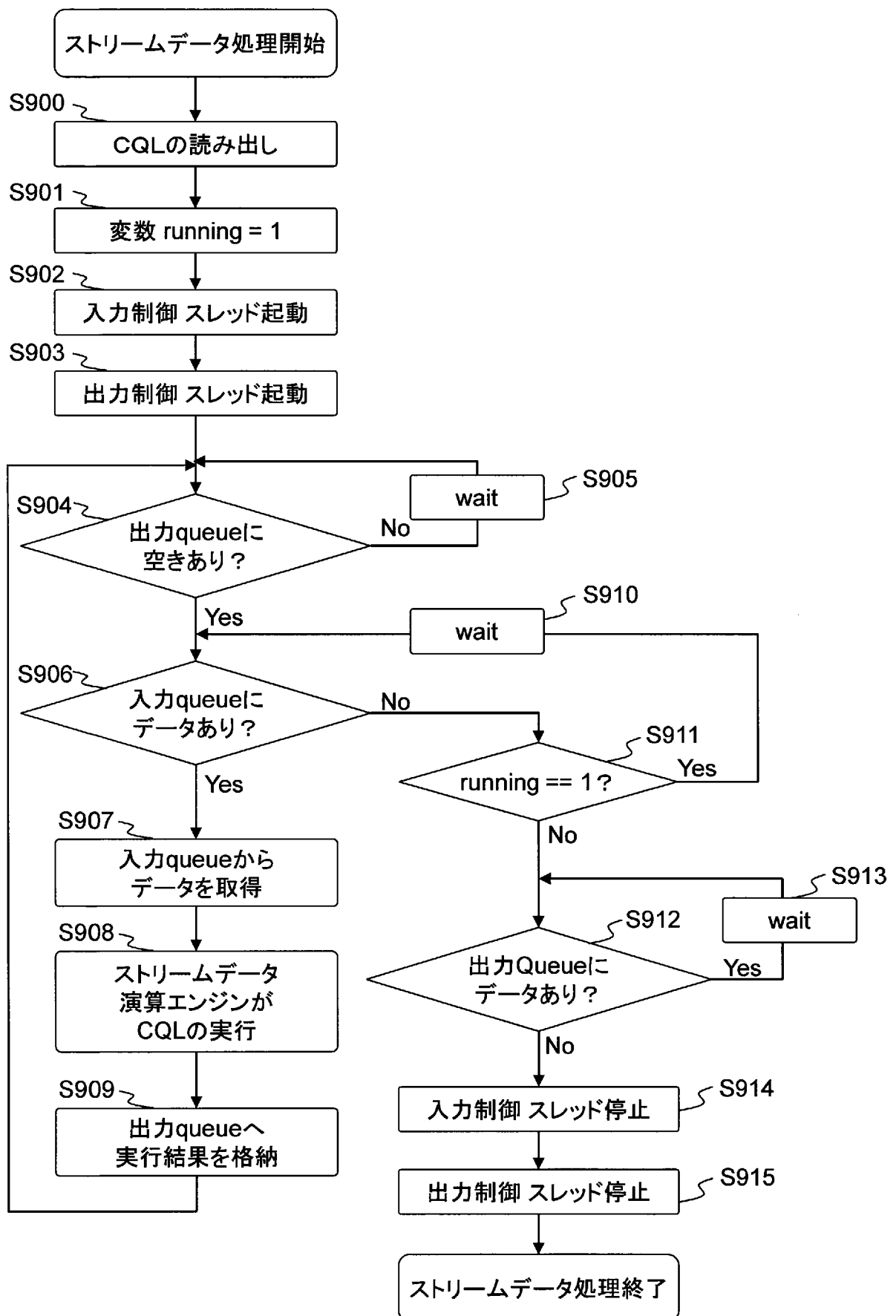


[図26]

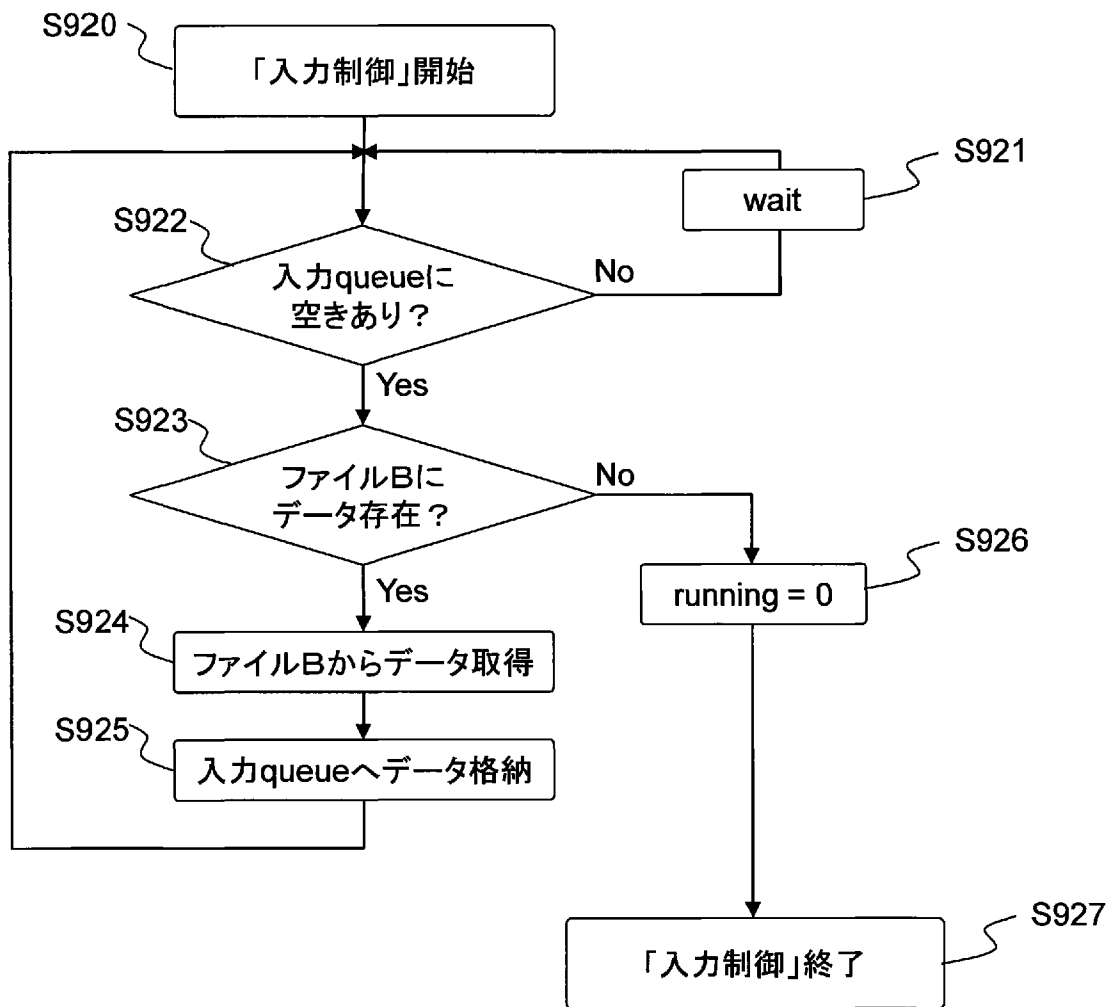


$$\frac{1}{2^i} \sum_{j=1}^{2^i} \max(a_{[i][j]} - a_{[i+1][j]}, a_{[i][j]} - a_{[i+1][j+1]}) < \text{閾値} \rightarrow \text{分割レベル} = i$$

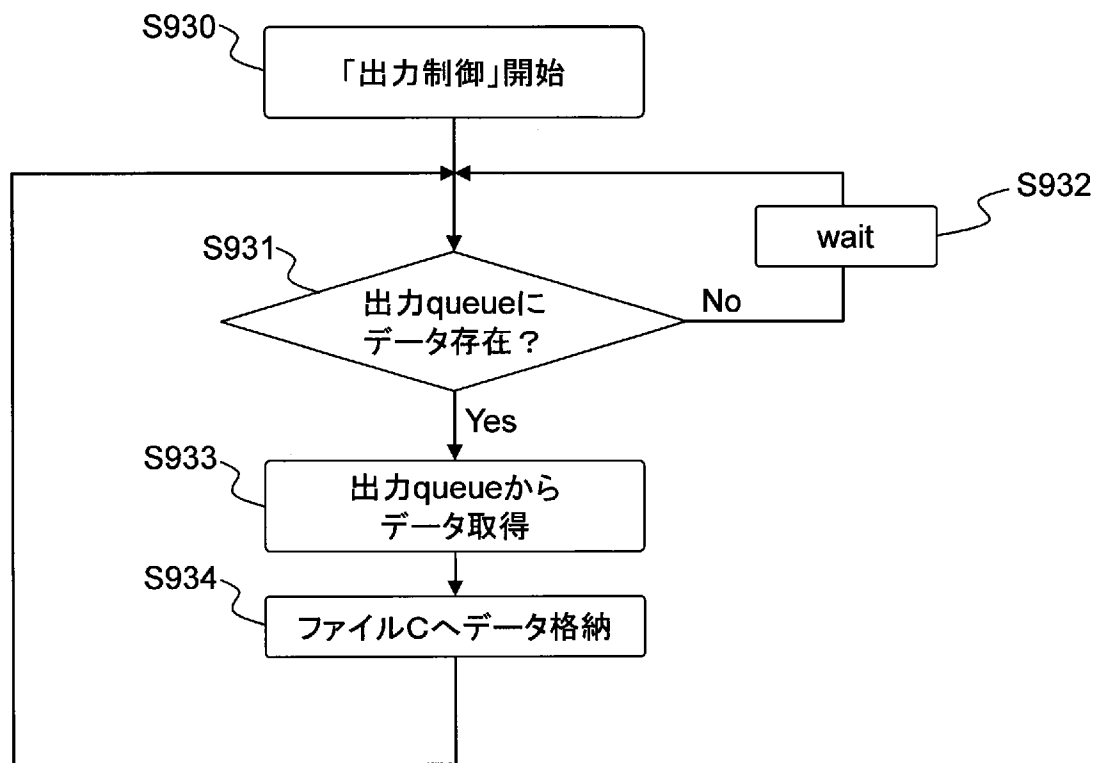
[図27]



[図28]



[図29]



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP2010/062343

A. CLASSIFICATION OF SUBJECT MATTER

G06F12/00 (2006.01) i, G06F17/30 (2006.01) i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F12/00, G06F17/30

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho 1922-1996 Jitsuyo Shinan Toroku Koho 1996-2010

Kokai Jitsuyo Shinan Koho 1971-2010 Toroku Jitsuyo Shinan Koho 1994-2010

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JP 2006-338432 A (Hitachi, Ltd.), 14 December 2006 (14.12.2006), entire text; all drawings & US 2006/0277230 A1 & US 2008/0256146 A1	1-28
A	JP 2008-310634 A (Mitsubishi Electric Corp.), 25 December 2008 (25.12.2008), abstract (Family: none)	1-28
A	JP 5-181720 A (Nippon Telegraph And Telephone Corp.), 23 July 1993 (23.07.1993), abstract (Family: none)	1-28

☐

Further documents are listed in the continuation of Box C.

☐

See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T"

later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X"

document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y"

document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&"

document member of the same patent family

Date of the actual completion of the international search

09 August, 2010 (09.08.10)

Date of mailing of the international search report

24 August, 2010 (24.08.10)

Name and mailing address of the ISA/
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

A. 発明の属する分野の分類（国際特許分類（I P C））

Int.Cl. G06F12/00(2006.01)i, G06F17/30(2006.01)i

B. 調査を行った分野

調査を行った最小限資料（国際特許分類（I P C））

Int.Cl. G06F12/00, G06F17/30

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報	1 9 2 2－1 9 9 6 年
日本国公開実用新案公報	1 9 7 1－2 0 1 0 年
日本国実用新案登録公報	1 9 9 6－2 0 1 0 年
日本国登録実用新案公報	1 9 9 4－2 0 1 0 年

国際調査で使用した電子データベース（データベースの名称、調査に使用した用語）

C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求項の番号
A	JP 2006-338432 A（株式会社日立製作所） 2006.12.14, 全文, 全図 & US 2006/0277230 A1 & US 2008/0256146 A1	1-28
A	JP 2008-310634 A（三菱電機株式会社） 2008.12.25, 要約（ファミリーなし）	1-28
A	JP 5-181720 A（日本電信電話株式会社） 1993.07.23, 要約（ファミリーなし）	1-28

C 欄の続きにも文献が列挙されている。

パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー

「A」特に関連のある文献ではなく、一般的技術水準を示すもの
「E」国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの
「L」優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献（理由を付す）
「O」口頭による開示、使用、展示等に言及する文献
「P」国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献

「T」国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの

「X」特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの

「Y」特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの

「&」同一パテントファミリー文献

国際調査を完了した日

0 9 . 0 8 . 2 0 1 0

国際調査報告の発送日

2 4 . 0 8 . 2 0 1 0

国際調査機関の名称及びあて先

日本国特許庁（I S A / J P）

郵便番号 1 0 0 - 8 9 1 5

東京都千代田区霞が関三丁目4番3号

特許庁審査官（権限のある職員）

桜井 茂行

5 M

2 9 4 5

電話番号 0 3 - 3 5 8 1 - 1 1 0 1 内線 3 5 9 9