(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
2015 年 2 月 5 日(05.02.2015)

WIPO | PCT

(10) 国際公開番号
WO 2015/015574 A1

(51) 国際特許分類:
*G06F 17/30* (2006.01) *G06F 9/50* (2006.01)

(21) 国際出願番号: PCT/JP2013/070664

(22) 国際出願日: 2013 年 7 月 30 日(30.07.2013)

(25) 国際出願の言語: 日本語

(26) 国際公開の言語: 日本語

(71) 出願人: 富士通株式会社 (FUJITSU LIMITED) [JP/JP]; 〒2118588 神奈川県川崎市中原区上小田中４丁目１番１号 Kanagawa (JP).

(72) 発明者: 小林 賢司 (KOBAYASHI, Kenji); 〒2118588 神奈川県川崎市中原区上小田中４丁目１番１号 富士通株式会社内 Kanagawa (JP). 佐伯敏章 (SAEKI, Toshiaki); 〒2118588 神奈川県川崎市中原区上小田中４丁目１番１号 富士通株式会社内 Kanagawa (JP).

(74) 代理人: 酒井 昭徳 (SAKAI, Akinori); 〒1006020 東京都千代田区霞が関３丁目２番５号 霞が関ビルディング２０階 酒井総合特許事務所 Tokyo (JP).

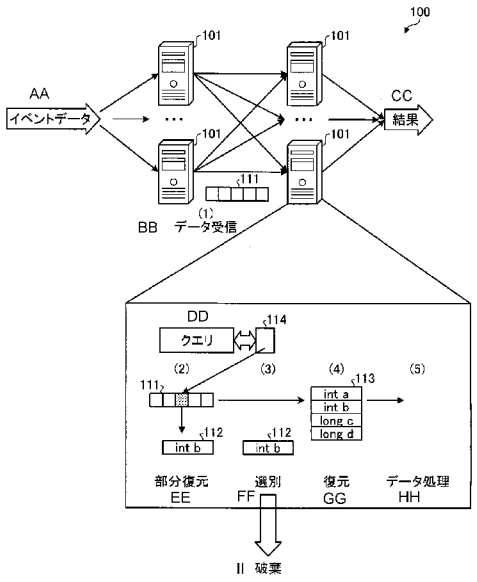
(81) 指定国 (表示のない限り、全ての種類の国内保護が可能): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, 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.

(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, RW, SD, SL, SZ, TZ, UG, ZM, ZW), ユーラシア (AM, AZ, BY, KG, KZ, 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, RS, SE, SI, SK, SM, TR), OAPI

[続葉有]

(54) **Title**: PROCESSING PROGRAM, PROCESSING SYSTEM, AND PROCESSING METHOD

(54) 発明の名称 : 処理プログラム、処理システムおよび処理方法




AA Event data
BB Data reception
CC Result
DD Query
EE Partial restoration
FF Selection
GG Restoration
HH Data processing
II Discarding

(57) **Abstract**: An engine node (101) receives serialized data (111) and deserializes a predetermined part (112) being referenced to select data on which a query is to be processed from the received data (111). The engine node (101) determines whether to discard the data (111) on the basis of the restored predetermined part (112) and conditions for selecting data on which the query is to be processed. If the engine node (101) determines that the data (111) is not to be discarded, the engine node (101) deserializes the received data (111). This enables the engine node (101) to avoid unnecessary deserialization.

(57) 要約: エンジンノード（１０１）は、シリアライズされたデータ（１１１）を受信し、受信したデータ（１１１）から、クエリの処理対象となるデータを選別するために参照される所定の部分（１１２）をデシリアライズする。エンジンノード（１０１）は、復元した所定の部分（１１２）とクエリが処理対象となるデータを選別するための条件とに基づいて、データ（１１１）を破棄するか否かを判断する。エンジンノード（１０１）は、破棄しないと判断した場合に、受信したデータ（１１１）をデシリアライズする。これにより、エンジンノード（１０１）は、不要なデシリアライズを削減する。

WO 2015/015574 A1

(BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, 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 ： 処理プログラム、処理システムおよび処理方法

## 技術分野

[0001] 本発明は、処理プログラム、処理システムおよび処理方法に関する。

## 背景技術

[0002] 従来、大量のイベントデータを複数のクエリで並列して処理し、最終結果を１つにして出力する分散並列型複合イベントプロセッシング（以下、分散並列型ＣＥＰ（Ｃｏｍｐｌｅｘ　Ｅｖｅｎｔ　Ｐｒｏｃｅｓｓｉｎｇ）と称する）と呼ばれる技術がある。複数のノードによる分散並列型ＣＥＰでは、クエリは複数のノード上に存在するため、ノード間での通信が発生する。

[0003] クエリの開発言語としては、開発効率を考慮したオブジェクト指向言語、例えば、Ｊａｖａ（登録商標）が採用されることが多い。構造化されたデータを送受信するには、送信側でデータの直列化処理、受信側で直列復元化処理が行われる。オブジェクト指向言語では、構造化データをオブジェクト、直列化処理をシリアライズ、直列化復元処理をデシリアライズと呼んでいる。

[0004] 分散並列型ＣＥＰでは大量のイベントデータが流れるため、各ノードにおけるシリアライズ、デシリアライズにかかる負荷は大きなものとなる。また、クエリは、例えば、前処理と後処理から構成され、前処理において、デシリアライズされたイベントデータを参照して、イベントデータの後処理を実行するか、イベントデータを破棄するかの判断が行われる。

[0005] 関連する先行技術としては、例えば、直列化された形式を保持するオブジェクトを用いた分散処理システムにおいて、イベントの通知のために、ネットワーク内にある特定のイベントの通知を受信するための装置を登録し、オブジェクトを登録要求とともに転送するものがある。

## 先行技術文献

## 特許文献

[0006] 特許文献1：特表２００２－５０５４７８号公報

## 発明の概要

### 発明が解決しようとする課題

[0007] しかしながら、従来技術によれば、クエリの前処理において破棄されるイベントデータであってもデシリアライズが行われ、ノードにかかる負荷の増大を招くという問題がある。例えば、分散並列型ＣＥＰでは、クエリの前処理において多くのイベントデータが破棄される傾向にあり、破棄されるイベントデータが多いにも拘わらず、すべてのイベントデータに対してデシリアライズが行われる。一方で、イベントデータをデシリアライズしなければ、クエリの前処理においてイベントデータを破棄するか否かを判断することができない。

[0008] 一つの側面では、本発明は、不要なデシリアライズを削減する処理プログラム、処理システムおよび処理方法を提供することを目的とする。

### 課題を解決するための手段

[0009] 本発明の一側面によれば、データを受信し、クエリの処理対象となるデータを選別するために参照される部分を特定する情報に基づいて、受信した前記データの所定の部分を復元し、復元した前記データの所定の部分と前記処理対象となるデータを選別するための条件とに基づいて、受信した前記データを破棄するか否かを判断し、破棄しないと判断した場合に、受信した前記データを復元する処理プログラム、処理システムおよび処理方法が提案される。

### 発明の効果

[0010] 本発明の一態様によれば、不要なデシリアライズを削減することができるという効果を奏する。

### 図面の簡単な説明

[0011] [図1]図１は、実施の形態１にかかる処理方法の一実施例を示す説明図である。

[図2]図２は、実施の形態１にかかるシステム２００のシステム構成例を示す説明図である。

[図3]図３は、エンジンノード１０１のハードウェア構成例を示すブロック図である。

[図4]図４は、データ１１１の一例を示す説明図である。

[図5]図５は、実施の形態１にかかるエンジンノード１０１の機能的構成例を示すブロック図である。

[図6]図６は、実施の形態１にかかるエンジンノード１０１のデータ処理手順の一例を示すフローチャートである。

[図7]図７は、実施の形態２にかかる処理方法の一実施例を示す説明図である。

[図8]図８は、実施の形態２にかかるマネージャノード２０１の機能的構成例を示すブロック図である。

[図9]図９は、実施の形態２にかかるコンパイラノード２０２の機能的構成例を示すブロック図である。

[図10]図１０は、実施の形態２にかかるエンジンノード１０１の機能的構成例を示すブロック図である。

[図11]図１１は、実施の形態２にかかるコンパイラノード２０２の復元コスト検証処理手順の一例を示すフローチャートである。

[図12]図１２は、実施の形態２にかかるエンジンノード１０１のデータ処理手順の一例を示すフローチャート（その１）である。

[図13]図１３は、実施の形態２にかかるエンジンノード１０１のデータ処理手順の一例を示すフローチャート（その２）である。

[図14]図１４は、実施の形態２にかかるエンジンノード１０１の透過率処理手順の一例を示すフローチャートである。

[図15]図１５は、実施の形態２にかかるマネージャノード２０１の部分復元判断処理手順の一例を示すフローチャートである。

[図16]図１６は、実施の形態３にかかる処理方法の一実施例を示す説明図で

ある。

[図17]図１７は、エンジンノードテーブル１７００の一例を示す説明図である。

[図18]図１８は、クエリテーブル１８００の一例を示す説明図である。

[図19]図１９は、Ｖｎｏｄｅ情報テーブル１９００の一例を示す説明図である。

[図20]図２０は、ルーティングテーブル２０００の一例を示す説明図である。

[図21]図２１は、実施の形態３にかかるエンジンノード１０１の透過率処理手順の一例を示すフローチャートである。

[図22]図２２は、実施の形態３にかかるマネージャノード２０１のＶｎｏｄｅ移動処理手順の一例を示すフローチャートである。

## 発明を実施するための形態

[0012] 以下に図面を参照して、本発明にかかる処理プログラム、処理システムおよび処理方法の実施の形態を詳細に説明する。

[0013] （実施の形態１）

図１は、実施の形態１にかかる処理方法の一実施例を示す説明図である。図１において、システム１００は、複数のエンジンノード１０１を含む分散並列型ＣＥＰシステムである。システム１００では、センサ等で発生したイベントデータがエンジンノード１０１に送信され、各エンジンノード１０１がイベントデータを処理し、最終結果が出力される。

[0014] エンジンノード１０１は、データ１１１を受信して、クエリを実行する情報処理装置である。クエリとは、予め定義されたルールに基づいて記述されたプログラムである。ルールは例えば、ＥＰＬ（Ｅｖｅｎｔ Ｐｒｏｃｅｓｓｉｎｇ Ｌａｎｇｕａｇｅ）等のクエリ記述言語で記述される。ルールをコンパイラで変換することによって、クエリが作成される。

[0015] 一方、クエリは開発効率性から開発言語として、Ｊａｖａのようなオブジェクト指向言語により記述されることが多い。オブジェクト指向言語は、オ

ブジェクトと呼ばれる機能の部品でソフトウェアを構成させるプログラミングに適した言語のことである。オブジェクトとは、現実世界に存在する物理的あるいは抽象的な実体を、属性と操作の集合としてモデル化し、コンピュータ上に再現したものである。また、オブジェクトには、Ｃ言語等の構造体のような構造化されたデータも含まれる。

[0016] データ１１１は、位置、温度、湿度等の環境情報および時間情報等を、科学的原理を応用して測定するセンサ等で発生したイベントデータを直列化したデータである。構造化されたデータを送受信するには、送信側でデータの直列化処理、受信側で直列復元化処理が行われる。オブジェクト指向言語では、構造化データをオブジェクト、直列化処理をシリアライズ、直列化復元処理をデシリアライズと呼んでいる。以下の説明では、クエリの開発言語の一例として、オブジェクト指向言語を例に挙げて説明する。

[0017] シリアライズとは、オブジェクト化されたイベントデータを、ネットワークを介して送受信するために、バイト列に直列化することである。デシリアライズとは、バイト列に直列化したデータをオブジェクト化されたイベントデータに復元することである。オブジェクト化されたイベントデータには、構造化されたイベントデータを含む。

[0018] すなわち、エンジンノード１０１が、他のエンジンノード１０１にオブジェクト化されたイベントデータを送信する際、イベントデータのシリアライズを実行してデータ１１１を作成し、他のエンジンノード１０１に送信する。シリアライズしたデータ１１１を受信した他のエンジンノード１０１は、データ１１１をデシリアライズし、オブジェクト化されたイベントデータを復元する。

[0019] 例えば、オブジェクト化されたイベントデータには、データ以外にデータ位置を示すポインタ等が含まれる。このため、エンジンノード１０１は、イベントデータの送信を行う際に、オブジェクト化されたイベントデータからポインタ等を削除して、データを直列に並べるシリアライズを行う。一方、エンジンノード１０１は、イベントデータを受信した際、シリアライズした

データ１１１にポインタ等を追加して、オブジェクト化されたイベントデータを復元するデシリアライズを行う。

[0020] ここで、クエリは、前処理と後処理から構成され、前処理において、デシリアライズされたイベントデータを参照して、イベントデータの後処理を実行するか、イベントデータを破棄するかの判断が行われる。イベントデータを破棄するか否かの判断は、デシリアライズされたイベントデータのすべてを参照して行われるのではなく、デシリアライズされたイベントデータの一部を参照して行われる。すなわち、データ１１１のすべてをデシリアライズすると、参照されない部分までデシリアライズされることになる。

[0021] そこで、実施の形態１では、エンジンノード１０１は、データ１１１を受信すると、データ１１１の所定の部分のみを復元し、復元した所定の部分を、クエリの処理対象を選別する条件と比較してデータ１１１を破棄するか判断する。そして、エンジンノード１０１は、データ１１１を破棄しないと判断した場合に、データ１１１のすべてを復元することにより、不要なデシリアライズを削減する。以下、実施の形態１にかかるエンジンノード１０１のデータ処理例について説明する。

[0022] （１）エンジンノード１０１は、クエリを実行するための直列化されたデータであるデータ１１１を受信する。

[0023] （２）エンジンノード１０１は、クエリの処理対象となるデータを選別するために参照される部分を特定する情報１１４に基づいて、受信したデータ１１１の所定の部分１１２をデシリアライズする。ここで、クエリの処理対象となるデータを選別するために参照される部分を特定する情報１１４とは、クエリごとに存在し、クエリにおいて、イベントデータを破棄するかの判断に使用する情報の位置を特定する情報である。

[0024] 図１の例では、クエリがデータを選別するための条件であるＥＰＬのフィルタ記述式が、『ｓｅｌｅｃｔ ａ ｆｒｏｍ ＥｖｅｎｔＡ ｗｈｅｒｅ ｂ＝１』と記述されているとする。この場合、情報１１４は、データ１１１の中でｂの位置を特定する情報である。エンジンノード１０１は、情報１

１４に基づいて、受信したデータ１１１から、所定の部分１１２をデシリアライズする。

[0025] （３）エンジンノード１０１は、復元した所定の部分１１２とクエリが処理対象となるデータを選別するための条件とに基づいて、データを破棄するか否かを判断する。データを選別するための条件とは、例えば、数値の一致・不一致、数値の大小、文字列の一致・不一致などを判断する演算子で構成される論理式である。

[0026] 図１の例では、クエリが処理対象となるデータを選別するための条件は、「ｂ＝１」である。従って、エンジンノード１０１は、復元した所定の部分１１２が「１」であるか否か確認する。所定の部分１１２が「１」でない受信したデータ１１１は、クエリのデータ選別後に使用されないデータである。この場合、エンジンノード１０１は、受信したデータ１１１を破棄する。

[0027] （４）エンジンノード１０１は、破棄しないと判断した場合に、受信したデータ１１１をデシリアライズする。エンジンノード１０１は、受信したデータ１１１をデシリアライズし、オブジェクト化されたイベントデータ１１３を復元する。図１の例では、エンジンノード１０１は、「ｉｎｔ　ａ」、「ｉｎｔ　ｂ」、「ｌｏｎｇ　ｃ」、「ｌｏｎｇ　ｄ」を含むデータ１１１をイベントデータ１１３にデシリアライズする。

[0028] （５）エンジンノード１０１は、デシリアライズしたイベントデータ１１３でクエリを実行し、実行結果を出力する。

[0029] 以上説明したように、エンジンノード１０１は、クエリを実行するための直列化されたデータ１１１を受信する。エンジンノード１０１は、クエリの処理対象となるデータを選別するために参照される部分１１２を特定する情報に基づいて、受信したデータ１１１の所定の部分１１２を復元する。エンジンノード１０１は、復元したデータ１１１の所定の部分１１２と処理対象となるデータを選別するための条件とに基づいて、受信したデータ１１１を破棄するか否かを判断し、破棄しないと判断した場合に、受信したデータ１１１を復元する。

[0030] これにより、クエリの後処理に渡されるデータ１１１に対してのみデータ全体のデシリアライズを行い、クエリの前処理において破棄されるデータ１１１に対してはデータ全体のデシリアライズを行わないようにすることができ、不要なデシリアライズを削減することができる。例えば、分散並列型ＣＥＰでは、多くのイベントデータは、所定の部分１１２を参照後に破棄されるため、不要なデシリアライズを大幅に削減することができる。

[0031]（システム２００のシステム構成例）

図２は、実施の形態１にかかるシステム２００のシステム構成例を示す説明図である。図２において、システム２００は、複数のエンジンノード１０１と、マネージャノード２０１と、コンパイラノード２０２と、を有する。エンジンノード１０１、マネージャノード２０１およびコンパイラノード２０２は、有線または無線のネットワーク２２０を介して接続される。ネットワーク２２０は、例えば、ＬＡＮ（Ｌｏｃａｌ Ａｒｅａ Ｎｅｔｗｏｒｋ）、ＷＡＮ（Ｗｉｄｅ Ａｒｅａ Ｎｅｔｗｏｒｋ）、インターネットなどである。

[0032] マネージャノード２０１は、システム２００全体を管理する制御装置である。コンパイラノード２０２は、ＥＰＬ等のクエリ記述言語で記述されたルールをコンパイラで変換することによって、クエリを作成するノードである。また、コンパイラノード２０２は、作成したクエリを各エンジンノード１０１に分配する機能を有する。マネージャノード２０１とコンパイラノード２０２は、図２では、別々のノードであるが、１つのノードとすることもできる。さらに、エンジンノード１０１と、マネージャノード２０１と、コンパイラノード２０２とを１つのノードとすることもできる。

[0033]（エンジンノード１０１のハードウェア構成例）

図３は、エンジンノード１０１のハードウェア構成例を示すブロック図である。図３において、エンジンノード１０１は、ＣＰＵ（Ｃｅｎｔｒａｌ Ｐｒｏｃｅｓｓｉｎｇ Ｕｎｉｔ）３０１と、メモリ３０２と、Ｉ／Ｆ（Ｉｎｔｅｒｆａｃｅ）３０３と、磁気ディスクドライブ３０４と、磁気ディス

ク３０５とを有する。また、各構成部は、バス３００によってそれぞれ接続される。

[0034] ここで、ＣＰＵ３０１は、エンジンノード１０１の全体の制御を司る。メモリ３０２は、例えば、ＲＯＭ（Ｒｅａｄ Ｏｎｌｙ Ｍｅｍｏｒｙ）、ＲＡＭ（Ｒａｎｄｏｍ Ａｃｃｅｓｓ Ｍｅｍｏｒｙ）およびフラッシュＲＯＭなどを有する。具体的には、例えば、フラッシュＲＯＭやＲＯＭが各種プログラムを記憶し、ＲＡＭがＣＰＵ３０１のワークエリアとして使用される。メモリ３０２に記憶されるプログラムは、ＣＰＵ３０１にロードされることで、コーディングされている処理をＣＰＵ３０１に実行させる。

[0035] Ｉ／Ｆ３０３は、通信回線を通じてネットワーク２２０に接続され、ネットワーク２２０を介して他のコンピュータ（例えば、図２に示したマネージャノード２０１およびコンパイラノード２０２）に接続される。そして、Ｉ／Ｆ３０３は、ネットワーク２２０と内部のインターフェースを司り、他のコンピュータからのデータの入出力を制御する。Ｉ／Ｆ３０３には、例えば、モデムやＬＡＮアダプタなどを採用することができる。

[0036] 磁気ディスクドライブ３０４は、ＣＰＵ３０１の制御にしたがって磁気ディスク３０５に対するデータのリード／ライトを制御する。磁気ディスク３０５は、磁気ディスクドライブ３０４の制御で書き込まれたデータを記憶する。

[0037] 上述した構成部のほか、例えば、ＳＳＤ（Ｓｏｌｉｄ Ｓｔａｔｅ Ｄｒｉｖｅ）、キーボード、マウス、ディスプレイなどを有することにしてもよい。また、図２に示したマネージャノード２０１およびコンパイラノード２０２についても、上述したエンジンノード１０１と同様のハードウェア構成例により実現することができる。

[0038] （データ１１１の一例）

図４は、データ１１１の一例を示す説明図である。図４に示す（１）は、データ１１１のデータ構造の一例であり、パラメータ長とパラメータデータが順に配置される。データ１１１のＬには、パラメータ長が格納され、この

後にパラメータデータが配置される。この場合、エンジンノード１０１は、クエリの処理対象となるデータを選別するために参照される部分を特定する情報として、パラメータの番号を有する。例えば、図４の例でｌｏｎｇ　ｃが、データを選別するために参照される部分である場合、３番目のデータが参照される部分を特定する情報となる。エンジンノード１０１は、先頭からパラメータ長（Ｌ）を読み、Ｌだけスキップして、再度パラメータ長（Ｌ）を読み、Ｌだけスキップして、目的とする３番目のパラメータに辿り着く。エンジンノード１０１は、目的のパラメータに辿り着いたら、ｌｏｎｇ　ｃを読み、受信したデータの所定の部分を復元する。

[0039] 図４に示す（２）は、データ１１１のデータ構造の他の例であり、クエリの処理対象となるデータを選別するために参照される部分は最初に配置される。エンジンノード１０１は、クエリの処理対象となるデータを選別するために参照される部分を特定する情報として、パラメータ長を有する。例えば、図４の例でｌｏｎｇ　ｃが、データを選別するために参照される部分である場合、ｌｏｎｇ　ｃのパラメータ長が参照される部分を特定する情報となる。エンジンノード１０１は、先頭からｌｏｎｇ　ｃを読み、受信したデータの所定の部分を復元する。

[0040] （エンジンノード１０１の機能的構成例）

図５は、実施の形態１にかかるエンジンノード１０１の機能的構成例を示すブロック図である。図５において、エンジンノード１０１は、受信部５０１と、復元部５０２と、選別部５０３と、処理部５０４と、変換部５０５と、送信部５０６とを含む構成である。各機能部は、具体的には、例えば、図３に示したメモリ３０２などの記憶装置に記憶されたプログラムをＣＰＵ３０１に実行させることにより、その機能を実現する。各機能部の処理結果は、例えば、図３に示したメモリ３０２などの記憶装置に記憶される。

[0041] 受信部５０１は、シリアライズしたデータ１１１を受信する機能を有する。例えば、受信部５０１は、センサ等から発生したイベントデータを集約するノードからデータ１１１を受信する。また、受信部５０１は、エンジンノ

ード１０１のクエリより前に実行されるクエリを実行するエンジンノード１０１からデータ１１１を受信する。受信部５０１は、受信したデータ１１１を復元部５０２に渡す。

[0042] 復元部５０２は、受信したデータ１１１をデシリアライズする機能を有する。復元部５０２は、クエリの処理対象となるデータを選別するために参照される部分を特定する情報に基づいて、受信したデータ１１１の所定の部分１１２をデシリアライズする。復元部５０２は、デシリアライズした所定の部分１１２を選別部５０３に渡す。また、復元部５０２は、受信したデータ１１１をすべてデシリアライズする。復元部５０２は、すべてデシリアライズしたイベントデータ１１３を処理部５０４に渡す。

[0043] 選別部５０３は、デシリアライズした所定の部分１１２と、クエリの処理対象となるデータを選別するための条件とに基づいて、受信したデータ１１１を破棄するか否かを判断する機能を有する。破棄しないと判断した場合、選別部５０３は、受信したデータ１１１を復元部５０２に渡す。破棄すると判断した場合、選別部５０３は、受信したデータ１１１を破棄する。

[0044] 選別部５０３は、クエリの処理対象となるデータを選別するための条件を記述する論理式に、所定の部分１１２の値を代入し、論理式が真であるとき、受信したデータ１１１を破棄せず、論理式が偽であるとき、受信したデータ１１１を破棄する。

[0045] 処理部５０４は、すべてデシリアライズしたイベントデータ１１３でクエリを実行する機能を有する。例えば、処理部５０４は、イベントデータ１１３のパラメータを加算する等の演算処理を実行する。また、処理部５０４は、イベントデータ１１３のパラメータが決められた値以上である時、アラームを発生させる処理等を実行する。処理部５０４は、クエリの実行により生成したデータを変換部５０５に渡す。

[0046] 変換部５０５は、処理部５０４が生成したデータをバイト列にシリアライズする機能を有する。変換部５０５は、シリアライズしたデータを送信部５０６に渡す。

[0047] 送信部５０６は、シリアライズしたデータを他のエンジンノード１０１に送信する機能を有する。

[0048] （エンジンノード１０１のデータ処理）

図６は、実施の形態１にかかるエンジンノード１０１のデータ処理手順の一例を示すフローチャートである。図６のフローチャートにおいて、まず、受信部５０１は、シリアライズしたデータ１１１を受信する（ステップＳ６０１）。次に、復元部５０２は、クエリの処理対象となるデータを選別するために参照される部分を特定する情報に基づいて、受信したデータ１１１の所定の部分１１２をデシリアライズする（ステップＳ６０２）。

[0049] 次に、選別部５０３は、デシリアライズした所定の部分１１２と、クエリの処理対象となるデータを選別するための条件とに基づいて、受信したデータ１１１を破棄するか否かを判断する（ステップＳ６０３）。破棄しないと判断した場合（ステップＳ６０３：Ｎｏ）、復元部５０２は、受信したデータ１１１をすべてデシリアライズする（ステップＳ６０５）。処理部５０４は、デシリアライズしたイベントデータ１１３でクエリを実行する（ステップＳ６０６）。

[0050] また、破棄すると判断した場合（ステップＳ６０３：Ｙｅｓ）、選別部５０３は、受信したデータ１１１を破棄する（ステップＳ６０４）。これにより、本フローチャートによる一連の処理は終了する。本フローチャートを実行することにより、選別部５０３で破棄されないデータ１１１のみがデシリアライズされる。

[0051] 以上説明したように、エンジンノード１０１は、クエリを実行するための直列化されたデータ１１１を受信する。エンジンノード１０１は、クエリの処理対象となるデータを選別するために参照される部分１１２を特定する情報に基づいて、受信したデータ１１１の所定の部分１１２を復元する。エンジンノード１０１は、復元したデータ１１１の所定の部分１１２と処理対象となるデータを選別するための条件とに基づいて、受信したデータ１１１を破棄するか否かを判断し、破棄しないと判断した場合に、受信したデータ１

１１を復元する。

[0052] これにより、実施の形態１にかかるエンジンノード１０１によれば、クエリの前処理において破棄されるデータ１１１に対してはデータ全体のデシリアライズを行わないようにして不要なデシリアライズを削減することができ、エンジンノード１０１にかかる負荷の増大を抑制することができる。

[0053] （実施の形態２）

次に、実施の形態２にかかる処理方法の一実施例について説明する。実施の形態２では、データ１１１の所定の部分１１２のデシリアライズ、およびデータ１１１のすべてのデシリアライズを行う条件が追加される。なお、実施の形態１で説明した箇所と同一箇所については、図示および説明を省略する。

[0054] 図７は、実施の形態２にかかる処理方法の一実施例を示す説明図である。図７において、エンジンノード１０１は、クエリの処理対象となるデータを選別するために参照される所定の部分１１２のデシリアライズの有効または無効を示すフラグ１２１を有する。エンジンノード１０１は、フラグ１２１が有効を示すときに、所定の部分１１２をデシリアライズする。マネージャノード２０１およびコンパイラノード２０２が、フラグ１２１の有効または無効を判断し、エンジンノード１０１に通知する。

[0055] エンジンノード１０１は、さらに、データ１１１の流用が可能または不可能を示すフラグ１２２を有する。エンジンノード１０１は、フラグ１２２が流用不可能を示すときに、データ１１１をデシリアライズする。コンパイラノード２０２は、データ１１１の流用が可能であるか否かを判断してエンジンノード１０１に通知する。なお、データ１１１の流用、所定の部分１１２のデシリアライズの有効または無効の判断、およびデータ１１１の流用が可能であるか否かの判断についての詳細な説明は後述する。

[0056] エンジンノード１０１は、データ１１１を受信した回数とデータ１１１を破棄しない回数を記憶する。エンジンノード１０１は、記憶したデータ１１１を受信した回数とデータ１１１を破棄しない回数に基づいて、クエリ透過

率を算出してマネージャノード２０１に送信する。クエリ透過率とは、データ１１１を受信した回数に対する、データ１１１を破棄しないと判断した回数の割合である。

[0057] マネージャノード２０１は、エンジンノード１０１からクエリ透過率を受信する。マネージャノード２０１は、受信したクエリ透過率に基づいて、所定の部分１１２のデシリアライズの有効または無効を判断し、判断結果をエンジンノード１０１に通知する。例えば、マネージャノード２０１は、クエリ透過率が所定値α以下である場合、所定の部分１１２のデシリアライズを有効と判断する。なお、所定値αについての詳細な説明は後述する。

[0058] コンパイラノード２０２は、ＥＰＬ等のクエリ記述言語を解析し、所定の部分１１２のデシリアライズの有効または無効を静的に判断する。コンパイラノード２０２は、判断結果をエンジンノード１０１に通知する。例えば、コンパイラノード２０２は、クエリの処理対象となるデータを選別するために参照される部分に含まれるパラメータが、所定の数以下であり、かつ、所定の型である場合に、受信したデータ１１１の所定の部分１１２をデシリアライズすると判断する。

[0059] ここで、所定の型とは、例えば、ｓｔｒｉｎｇ、ｃｌａｓｓのように内部にパラメータを有する型である。また、コンパイラノード２０２は、クエリの処理対象となるデータを選別するために参照される部分をデシリアライズするコストを算出する。コンパイラノード２０２は、算出したコストがデータ１１１をすべてデシリアライズするコスト未満である場合に、受信したデータ１１１の所定の部分１１２をデシリアライズすると判断することもできる。ここで、コストとは、エンジンノード１０１がデシリアライズするために使用するＣＰＵ時間、またはメモリのことである。

[0060] また、コンパイラノード２０２は、ＥＰＬ等のクエリ記述言語を解析し、エンジンノード１０１のクエリを実行した後に実行される他のエンジンノード１０１のクエリに参照される情報がデータ１１１に含まれ、データ１１１の流用が可能か否かを判断する。コンパイラノード２０２は、判断結果をエ

ンジンノード１０１に通知する。以下、実施の形態２にかかるエンジンノード１０１のデータ処理例について説明する。

[0061] （１）エンジンノード１０１は、シリアライズされたデータ１１１を受信する。エンジンノード１０１は、クエリを実行するための直列化されたデータであるデータ１１１を受信する。

[0062] （２）エンジンノード１０１は、所定の部分１１２のデシリアライズが有効か無効か判断する。エンジンノード１０１は、マネージャノード２０１およびコンパイラノード２０２から通知された所定の部分１１２のデシリアライズの有効または無効を示すフラグ１２１を参照し、所定の部分１１２のデシリアライズが有効か無効か判断する。

[0063] （３）エンジンノード１０１は、データ１１１を受信した回数を保存する。エンジンノード１０１は、クエリ透過率を計算するため、データ１１１を受信した回数を保存する。

[0064] （４）エンジンノード１０１は、所定の部分１１２のデシリアライズが有効である場合、受信したデータ１１１から、クエリの処理対象となるデータを選別するために参照される所定の部分１１２をデシリアライズする。図７の例では、図１と同様にエンジンノード１０１は、ｂの部分を特定する情報に基づいて、受信したデータ１１１から、所定の部分１１２をデシリアライズする。また、無効である場合、エンジンノード１０１は、受信したデータ１１１をすべてデシリアライズする。

[0065] （５）エンジンノード１０１は、復元した所定の部分１１２とクエリが処理対象となるデータを選別するための条件とに基づいて、データ１１１を破棄するか否かを判断する。エンジンノード１０１は、データ１１１を破棄しない回数を記憶する。図７の例では、図１と同様に、エンジンノード１０１は、復元した所定の部分１１２が１であるか否か確認する。所定の部分１１２が１でない受信したデータ１１１は、クエリのデータ選別後に使用されないデータである。この場合、エンジンノード１０１は、受信したデータ１１１を破棄する。

[0066] （６）エンジンノード１０１は、データ１１１を破棄しないと判断した回数を、データ１１１を受信した回数で割ることによりクエリ透過率を計算し、マネージャノード２０１に通知する。マネージャノード２０１が所定の部分１１２のデシリアライズの有効または無効を動的に判断する情報として、エンジンノード１０１は、クエリ透過率をマネージャノード２０１に通知する。

[0067] （７）エンジンノード１０１は、データ１１１の流用が可能か否かを判断する。コンパイラノード２０２から通知されたデータ１１１の流用の可能または不可能を示すフラグ１２２を参照し、データ１１１の流用が可能か否かを判断する。

[0068] （８）エンジンノード１０１は、流用が可能と判断した場合、受信したデータ１１１のデシリアライズを行わない。

[0069] （９）エンジンノード１０１は、受信したデータ１１１でクエリを実行する。エンジンノード１０１は、データ１１１でクエリを実行し、結果を出力する。

[0070] 以上説明したように、エンジンノード１０１は、クエリの処理対象となるデータを選別するために参照される部分に含まれるパラメータが、所定の数以下であり、かつ、所定の型である場合に、データ１１１の所定の部分１１２をデシリアライズする。また、エンジンノード１０１は、クエリの処理対象となるデータを選別するために参照される部分をデシリアライズするコストが、データ１１１をすべてデシリアライズするコスト未満である場合に、データ１１１の所定の部分１１２をデシリアライズする。さらに、エンジンノード１０１は、クエリ透過率が所定値αα以下である場合に、データ１１１の所定の部分１１２をデシリアライズする。

[0071] これにより、不要なデシリアライズを削減する効果が見込まれる場合に、エンジンノード１０１は、所定の部分１１２をデシリアライズすることができる。

[0072] また、エンジンノード１０１は、データ１１１を破棄しないと判断し、か

つ、クエリを実行した後に実行される他のクエリに参照される情報がデータ１１１に含まれる場合は、データ１１１を復元しない。

[0073] これにより、エンジンノード１０１は、エンジンノード１０１のクエリを実行した後に実行される他のエンジンノード１０１までデシリアライズを行わないことが可能になる。このため、エンジンノード１０１は、デシリアライズを削減することができる。

[0074] （マネージャノード２０１の機能的構成例）

図８は、実施の形態２にかかるマネージャノード２０１の機能的構成例を示すブロック図である。図８において、マネージャノード２０１は、受信部８０１と、管理部８０２と、通知部８０３とを含む構成である。各機能部は、具体的には、例えば、図３に示したメモリ３０２などの記憶装置に記憶されたプログラムをＣＰＵ３０１に実行させることにより、その機能を実現する。各機能部の処理結果は、例えば、図３に示したメモリ３０２などの記憶装置に記憶される。

[0075] 受信部８０１は、エンジンノード１０１からクエリ透過率を受信する機能を有する。受信部８０１は、受信したクエリ透過率を管理部８０２に渡す。

[0076] 管理部８０２は、受信したクエリ透過率から所定の部分１１２のデシリアライズの有効または無効を判断する機能を有する。管理部８０２は、判断した判断結果を通知部８０３に渡す。例えば、管理部８０２は、クエリ透過率が所定値α以下である場合、所定の部分１１２のデシリアライズを有効と判断する。

[0077] 通知部８０３は、判断結果をエンジンノード１０１に通知する機能を有する。具体的には、例えば、通知部８０３は、管理部８０２が判断した判断結果を、受信部８０１が受信したクエリ透過率の送信元のエンジンノード１０１に通知する。

[0078] なお、上述した所定値αは、予め設定されていてもよく、例えば、マネージャノード２０１が以下のように決定することにしてもよい。

[0079] ここで、所定の部分１１２をデシリアライズした場合のデシリアライズコ

ストは、下記式（１）を用いて表すことができる。また、所定の部分１１２をデシリアライズしない場合のデシリアライズコストは、下記式（２）を用いて表すことができる。ただし、Ｃ１は、所定の部分１１２をデシリアライズした場合のデシリアライズコストである。Ｃ２は、所定の部分１１２をデシリアライズしない場合のデシリアライズコストである。ｃ１は、所定の部分１１２をデシリアライズするコストである。ｃ２は、データ１１１をすべてデシリアライズするコストである。ｐは、クエリ透過率である。

[0080] Ｃ１＝ｃ１＋ｃ２×ｐ ・・・（１）

[0081] Ｃ２＝ｃ２ ・・・（２）

[0082] デシリアライズコストＣ１は、クエリ透過率ｐがある値より大きくなると、デシリアライズコストＣ２を超える。すなわち、クエリ透過率ｐがある値より大きいと、所定の部分１１２をデシリアライズする場合は、所定の部分１１２をデシリアライズしない場合と比べてデシリアライズコストが増加する。このため、マネージャノード２０１は、例えば、デシリアライズコストＣ１がデシリアライズＣ２より小さくなるクエリ透過率ｐの値を所定値αに決定する。

[0083]（コンパイラノード２０２の機能的構成例）

図９は、実施の形態２にかかるコンパイラノード２０２の機能的構成例を示すブロック図である。図９において、コンパイラノード２０２は、構文解析部９０１と、復元コスト検証部９０２と、分配部９０３とを含む構成である。各機能部は、具体的には、例えば、図３に示したメモリ３０２などの記憶装置に記憶されたプログラムをＣＰＵ３０１に実行させることにより、その機能を実現する。各機能部の処理結果は、例えば、図３に示したメモリ３０２などの記憶装置に記憶される。

[0084] 構文解析部９０１は、ＥＰＬ等のクエリ記述言語を解析し、実行可能なクエリに変換する機能を有する。構文解析部９０１は、変換したクエリを分配部９０３に渡す。

[0085] 復元コスト検証部９０２は、ＥＰＬ等のクエリ記述言語を解析し、所定の

部分１１２のデシリアライズの有効または無効を判断する機能を有する。復元コスト検証部９０２は、有効または無効の判断結果を分配部９０３に渡す。復元コスト検証部９０２は、例えば、クエリの処理対象となるデータを選別するために参照される部分に含まれるパラメータが、所定の数以下であり、かつ、所定の型である場合に、受信したデータ１１１の所定の部分１１２をデシリアライズすると判断する。また、復元コスト検証部９０２は、クエリの処理対象となるデータを選別するために参照される部分をデシリアライズするコストを算出する。復元コスト検証部９０２は、算出したコストがデータ１１１をすべてデシリアライズするコスト未満である場合に、受信したデータ１１１の所定の部分１１２をデシリアライズすると判断することもできる。

[0086] 復元コスト検証部９０２は、さらに、ＥＰＬ等のクエリ記述言語を解析し、データ１１１の流用が可能か否かを判断する機能を有する。復元コスト検証部９０２は、データ１１１の流用可能かの判断結果を分配部９０３に渡す。例えば、エンジンノード１０１のクエリのＥＰＬが『ｉｎｓｅｒｔ　ＥｖｅｎｔＢ　ｓｅｌｅｃｔ　ａ，ｃ，ｄ，ｅ，ｆ　ｆｒｏｍ　ＥｖｅｎｔＡ　ｗｈｅｒｅ　ｂ＝１００』で記述されているとする。

[0087] また、当該クエリを実行した後に、後のエンジンノード１０１で実行されるクエリのＥＰＬが、『ｓｅｌｅｃｔ　ａ，ｃ　ｆｒｏｍ　ＥｖｅｎｔＢ　ｗｈｅｒｅ　ｂ＝１００　ａｎｄ　ｅ＝”ＡＡＡ”』で記述されているとする。この場合、エンジンノード１０１のイベントデータであるＥｖｅｎｔＡのパラメータは、そのまま後のエンジンノード１０１のイベントデータであるＥｖｅｎｔＢのパラメータとして、使用されている。このため、コンパイラノード２０２はデータ流用が可能であると判断する。

[0088] これに対して、エンジンノード１０１のクエリのＥＰＬが、『ｉｎｓｅｒｔ　ＥｖｅｎｔＢ　ｓｅｌｅｃｔ　ａ，ｂ，ｃｏｕｎｔ（ａ），ｓｕｍ（ｂ）　ｆｒｏｍ　ＥｖｅｎｔＡ　ｗｈｅｒｅ　ｂ＝１００』で記述されているとする。この場合、後のエンジンノード１０１のイベントデータであるＥｖ

ｅｎｔＢのパラメータは、エンジンノード１０１のイベントデータであるＥｖｅｎｔＡのパラメータに、新しいパラメータを追加したパラメータである。このため、コンパイラノード２０２はデータ流用が不可能であると判断する。

[0089] 分配部９０３は、変換したクエリ、所定の部分１１２のデシリアライズの有効または無効の判断結果、およびデータ１１１の流用可能かの判断結果をエンジンノード１０１に分配する機能を有する。

[0090] （エンジンノード１０１の機能的構成例）

図１０は、実施の形態２にかかるエンジンノード１０１の機能的構成例を示すブロック図である。図１０において、エンジンノード１０１は、受信部５０１と、復元部５０２と、選別部５０３と、処理部５０４と、変換部５０５と、送信部５０６と、制御部１００１と、を含む構成である。制御部１００１は、具体的には、例えば、図３に示したメモリ３０２などの記憶装置に記憶されたプログラムをＣＰＵ３０１に実行させることにより、その機能を実現する。各機能部の処理結果は、例えば、図３に示したメモリ３０２などの記憶装置に記憶される。

[0091] 受信部５０１は、シリアライズしたデータ１１１を受信した場合、シリアライズしたデータ１１１を受信したことを制御部１００１に通知する機能をさらに有する。

[0092] 復元部５０２は、所定の部分１１２のデシリアライズの有効または無効を示すフラグ１２１を参照する機能をさらに有する。復元部５０２は、所定の部分１１２のデシリアライズが有効である場合、クエリの処理対象となるデータを選別するために参照される部分を特定する情報に基づいて、受信したデータ１１１の所定の部分１１２をデシリアライズする。

[0093] 選別部５０３は、データ１１１を破棄しないと判断した場合、データ１１１を破棄しないことを制御部１００１に通知する機能をさらに有する。

[0094] 処理部５０４は、コンパイラノード２０２から通知されたデータ１１１の流用の可能または不可能を示すフラグ１２２を参照する機能をさらに有する

。処理部５０４は、データ１１１の流用が可能な場合、受信したデータ１１１のデシリアライズを行わない。

[0095] 制御部１００１は、データ１１１を破棄しないと判断した回数を、データ１１１を受信した回数で割ることによりクエリ透過率を計算し、マネージャノード２０１に通知する機能を有する。また、制御部１００１は、マネージャノード２０１およびコンパイラノード２０２の通知に基づいて、所定の部分１１２のデシリアライズの有効または無効を示すフラグ１２１を更新する機能をさらに有する。さらに、制御部１００１は、コンパイラノード２０２の通知に基づいて、データ１１１の流用が可能または不可能を示すフラグ１２２を更新する機能をさらに有する。

[0096] （コンパイラノード２０２の復元コスト検証処理）

図１１は、実施の形態２にかかるコンパイラノード２０２の復元コスト検証処理手順の一例を示すフローチャートである。図１１のフローチャートにおいて、まず、復元コスト検証部９０２は、ＥＰＬ等のクエリ記述言語を解析する（ステップＳ１１０１）。次に、復元コスト検証部９０２は、クエリの処理対象となるデータを選別するために参照される部分に含まれるパラメータが、所定の数以下であり、かつ、所定の型であるか否かを確認する（ステップＳ１１０２）。

[0097] パラメータが、所定の数以下であり、かつ、所定の型である場合に（ステップＳ１１０２：Ｙｅｓ）、復元コスト検証部９０２は、所定の部分１１２のデシリアライズを有効と判断する（ステップＳ１１０３）。パラメータが、所定の数より多い、または、所定の型でない場合に（ステップＳ１１０２：Ｎｏ）、復元コスト検証部９０２は、所定の部分１１２のデシリアライズを無効と判断する（ステップＳ１１０４）。

[0098] 次に、復元コスト検証部９０２は、ＥＰＬ等のクエリ記述言語を解析し、データ１１１の流用が可能か否かを判断する（ステップＳ１１０５）。データ流用が可能であると判断した場合（ステップＳ１１０５：Ｙｅｓ）、復元コスト検証部９０２は、データ１１１の流用転送を可能と判断する（ステッ

プＳ１１０６）。データ流用が不可能であると判断した場合（ステップＳ１１０５：Ｎｏ）、復元コスト検証部９０２は、データ１１１の流用転送を不可能と判断する（ステップＳ１１０７）。これにより、本フローチャートによる一連の処理は終了する。本フローチャートを実行することにより、所定の部分１１２のデシリアライズの有効、無効、およびデータ１１１の流用転送可能、不可能が判断される。

[0099] （エンジンノード１０１のデータ処理）

図１２は、実施の形態２にかかるエンジンノード１０１のデータ処理手順の一例を示すフローチャート（その１）である。図１２のフローチャートにおいて、まず、受信部５０１は、シリアライズしたデータ１１１を受信する（ステップＳ１２０１）。次に、復元部５０２は、デシリアライズの有効または無効を示すフラグ１２１を参照し、所定の部分１１２のデシリアライズの有効または無効を確認する（ステップＳ１２０２）。所定の部分１１２のデシリアライズが有効である場合（ステップＳ１２０２：Ｙｅｓ）、復元部５０２は、所定の部分１１２をデシリアライズする（ステップＳ１２０３）。所定の部分１１２のデシリアライズが無効である場合（ステップＳ１２０２：Ｎｏ）、復元部５０２は、受信したデータ１１１をすべてデシリアライズする（ステップＳ１２０４）。

[0100] 受信部５０１は、シリアライズしたデータ１１１を受信したことを制御部１００１に通知する（ステップＳ１２０５）。次に、選別部５０３は、デシリアライズした所定の部分１１２と、クエリの処理対象となるデータを選別するための条件とに基づいて、受信したデータ１１１を破棄するか否かを判断する（ステップＳ１２０６）。破棄しないと判断した場合（ステップＳ１２０６：Ｎｏ）、選別部５０３は、受信したデータ１１１を処理部５０４に転送し（ステップＳ１２０８）、データ１１１を破棄しないことを制御部１００１に通知する（ステップＳ１２０９）。

[0101] また、破棄すると判断した場合（ステップＳ１２０６：Ｙｅｓ）、選別部５０３は、受信したデータ１１１を破棄する（ステップＳ１２０７）。これ

により、本フローチャートによる一連の処理は終了する。本フローチャートを実行することにより、所定の部分１１２のデシリアライズが有効の場合に、所定の部分１１２がデシリアライズされ、選別部５０３で破棄されないデータ１１１のみがデシリアライズされる。

[0102] 図１３は、実施の形態２にかかるエンジンノード１０１のデータ処理手順の一例を示すフローチャート（その２）である。図１３のフローチャートにおいて、まず、処理部５０４は、デシリアライズの有効または無効を示すフラグ１２１を参照し、所定の部分１１２のデシリアライズの有効または無効を確認する（ステップＳ１３０１）。所定の部分１１２のデシリアライズが有効である場合（ステップＳ１３０１：Ｙｅｓ）、処理部５０４は、データ１１１の流用の可能または不可能を示すフラグ１２２を参照し、データ１１１の流用が可能か否かを確認する（ステップＳ１３０２）。

[0103] 流用が可能な場合（ステップＳ１３０２：Ｙｅｓ）、処理部５０４は、受信したデータ１１１でクエリを実行する（ステップＳ１３０４）。この後、ステップＳ１３０８に移行する。一方、データ１１１の流用が不可能な場合（ステップＳ１３０２：Ｎｏ）、復元部５０２は、受信したデータ１１１をすべてデシリアライズする（ステップＳ１３０３）。この後、ステップＳ１３０５に移行する。

[0104] 所定の部分１１２のデシリアライズが無効である場合（ステップＳ１３０１：Ｎｏ）、処理部５０４は、デシリアライズしたイベントデータ１１３でクエリを実行する（ステップＳ１３０５）。処理部５０４は、データ処理で生成した新しいパラメータをデータに追加して新データを作成する（ステップＳ１３０６）。変換部５０５は、新データをバイト列にシリアライズする（ステップＳ１３０７）。

[0105] 最後に、送信部５０６は、シリアライズしたデータを他のエンジンノード１０１に送信する（ステップＳ１３０８）。これにより、本フローチャートによる一連の処理は終了する。本フローチャートを実行することにより、データ１１１の流用が可能な場合、クエリの選別処理で通過したデータ１１１

はデシリアライズされない。

[0106] （エンジンノード１０１の透過率処理）

図１４は、実施の形態２にかかるエンジンノード１０１の透過率処理手順の一例を示すフローチャートである。図１４のフローチャートにおいて、まず、制御部１００１は、データ１１１を破棄しないと判断した回数とデータ１１１を受信した回数からクエリ透過率を計算する（ステップＳ１４０１）。次に、制御部１００１は、計算したクエリ透過率をマネージャノード２０１に通知する（ステップＳ１４０２）。これにより、本フローチャートによる一連の処理は終了する。本フローチャートを実行することにより、クエリ透過率がマネージャノード２０１に通知される。

[0107] （マネージャノード２０１の部分復元判断処理）

図１５は、実施の形態２にかかるマネージャノード２０１の部分復元判断処理手順の一例を示すフローチャートである。図１５のフローチャートにおいて、まず、受信部８０１は、エンジンノード１０１からクエリ透過率を受信する（ステップＳ１５０１）。次に、管理部８０２は、クエリ透過率が所定値より大きいかどうか確認する（ステップＳ１５０２）。

[0108] クエリ透過率が所定値より大きい場合（ステップＳ１５０２：Ｙｅｓ）、管理部８０２は、所定の部分１１２のデシリアライズを無効と判断する（ステップＳ１５０３）。クエリ透過率が所定値以下である場合（ステップＳ１５０２：Ｎｏ）、管理部８０２は、所定の部分１１２のデシリアライズを有効と判断する（ステップＳ１５０４）。これにより、本フローチャートによる一連の処理は終了する。本フローチャートを実行することにより、所定の部分１１２のデシリアライズの有効、無効が判断される。

[0109] 以上説明したように、エンジンノード１０１は、クエリの処理対象となるデータを選別するために参照される部分に含まれるパラメータが、所定の数以下であり、かつ、所定の型である場合に、データ１１１の所定の部分１１２をデシリアライズする。また、エンジンノード１０１は、クエリの処理対象となるデータを選別するために参照される部分をデシリアライズするコス

トが、データ１１１をすべてデシリアライズするコスト未満である場合に、データ１１１の所定の部分１１２をデシリアライズする。さらに、エンジンノード１０１は、透過率が所定値以下である場合に、データ１１１の所定の部分１１２をデシリアライズする。

[0110] これにより、不要なデシリアライズを削減する効果が見込まれる場合に、エンジンノード１０１は、所定の部分１１２をデシリアライズすることができる。

[0111] また、エンジンノード１０１は、データ１１１を破棄しないと判断し、かつ、クエリを実行した後に実行される他のクエリに参照される情報がデータ１１１に含まれる場合は、データ１１１を復元しない。

[0112] これにより、エンジンノード１０１は、エンジンノード１０１のクエリを実行した後に実行される他のエンジンノード１０１までデシリアライズを行わないことが可能になる。このため、エンジンノード１０１は、デシリアライズを削減することができる。

[0113] （実施の形態３）

次に、実施の形態３にかかる処理方法の一実施例について説明する。実施の形態３では、クエリを実行するための直列化されたデータをまとめたデータ群単位のクエリ透過率を判断基準にして、データ群の割当先のエンジンノード１０１を決定する場合について説明する。なお、実施の形態１および実施の形態２で説明した箇所と同一箇所については、図示および説明を省略する。

[0114] 図１６は、実施の形態３にかかる処理方法の一実施例を示す説明図である。実施の形態３において、同一クエリを並列処理する際、クエリを実行するための直列化された複数のデータは、複数のデータ群に分割される。マネージャノード２０１は、データ群ごとにデータをエンジンノード１０１に割り当てる。

[0115] 例えば、シリアライズされたデータにはｉｄ（パラメータ）が付与されており、複数のデータは、各データのｉｄに従って、指定のパラメータ値範囲

で複数のデータ群に分割される。なお、ｉｄは、シリアライズされたデータの識別子である。ここでは、あるパラメータ値範囲のデータ群を「Ｖｎｏｄｅ」と表記する。

[0116] 図１６の例では、例えば、Ｖｎｏｄｅ１６０１は、ｉｄが「０００～０９９」のデータ群であり、エンジンノード１０１－１に割り当てられている。また、例えば、Ｖｎｏｄｅ１６０４は、ｉｄが「３００～３９９」のデータ群であり、エンジンノード１０１－２に割り当てられている。

[0117] エンジンノード１０１は、Ｖｎｏｄｅごとにデータ１１１を受信した回数とデータ１１１を破棄しない回数を記憶する。そして、エンジンノード１０１は、データ１１１を受信した回数に対する、データ１１１を破棄しないと判断した回数の割合であるＶｎｏｄｅごとのクエリ透過率を算出して、マネージャノード２０１に送信する。

[0118] マネージャノード２０１は、エンジンノード１０１からＶｎｏｄｅごとのクエリ透過率を受信する。マネージャノード２０１は、エンジンノード１０１の負荷が規定値より高くなった場合、例えば、エンジンノード１０１のＣＰＵ使用率またはメモリ使用率が規定値より高くなった場合、Ｖｎｏｄｅを他のエンジンノード１０１に割り当てる。この際、マネージャノード２０１は、エンジンノード１０１全体のクエリ透過率とＶｎｏｄｅごとのクエリ透過率に基づいて、移動するＶｎｏｄｅと割当先のエンジンノード１０１を決定する。

[0119] 例えば、マネージャノード２０１は、所定の部分１１２のデシリアライズの有効であるエンジンノード１０１、すなわちクエリ透過率が所定値$\alpha$以下となるエンジンノード１０１に、クエリ透過率が所定値$\beta$以下となるＶｎｏｄｅを割り当てる。また、マネージャノード２０１は、所定の部分１１２のデシリアライズの無効であるエンジンノード１０１、すなわちクエリ透過率が所定値$\alpha$より大きいエンジンノード１０１に、クエリ透過率が所定値$\gamma$以上のＶｎｏｄｅを割り当てる。所定値$\beta$は、所定値$\alpha$以下の値である。また、所定値$\gamma$は、所定値$\alpha$以上の値である。

[0120] 図１６の例で、エンジンノード１０１－１が所定の部分１１２のデシリアライズが無効であり、エンジンノード１０１－２が所定の部分１１２のデシリアライズが有効であるとする。また、エンジンノード１０１－１において、Ｖｎｏｄｅ１６０１のクエリ透過率が所定値β以下であり、Ｖｎｏｄｅ１６０２のクエリ透過率が所定値γ以上とする。また、エンジンノード１０１－２において、Ｖｎｏｄｅ１６０４のクエリ透過率が所定値β以下であり、Ｖｎｏｄｅ１６０５のクエリ透過率が所定値γ以上とする。

[0121] この場合、エンジンノード１０１－１の負荷が規定値より高くなり、エンジンノード１０１－１のＶｎｏｄｅをエンジンノード１０１－２に割り当てる場合、マネージャノード２０１は、Ｖｎｏｄｅ１６０２をエンジンノード１０１－２に割り当てる。一方、エンジンノード１０１－２の負荷が規定値より高くなり、エンジンノード１０１－２のＶｎｏｄｅをエンジンノード１０１－１に割り当てる場合、マネージャノード２０１は、Ｖｎｏｄｅ１６０４をエンジンノード１０１－１に割り当てる。

[0122] これにより、マネージャノード２０１は、クエリ透過率が所定値β以下のＶｎｏｄｅを、所定の部分１１２のデシリアライズが有効であるエンジンノード１０１に割り当てることができる。また、マネージャノード２０１は、クエリ透過率が所定値γ以上のＶｎｏｄｅを、所定の部分１１２のデシリアライズが無効であるエンジンノード１０１に割り当てることができる。このため、マネージャノード２０１は、エンジンノード１０１での不要なデシリアライズの削減を考慮した負荷均衡化を行うことができる。

[0123] （エンジンノードテーブル１７００の一例）

図１７は、エンジンノードテーブル１７００の一例を示す説明図である。図１７において、エンジンノードテーブル１７００は、エンジンノードＩＤ、ＣＰＵ使用率およびメモリ使用率のフィールドを有し、各フィールドに情報を設定することで、エンジンノード情報（例えば、エンジンノード情報１７００－１～１７００－３）をレコードとして記憶する。

[0124] ここで、エンジンノードＩＤは、エンジンノード１０１の識別子である。

ＣＰＵ使用率は、エンジンノード１０１のＣＰＵの使用率である。メモリ使用率は、エンジンノード１０１のメモリの使用率である。例えば、エンジンノード情報１７００－１は、エンジンノード１０１のエンジンノードＩＤ「１」、ＣＰＵ使用率「６２％」およびメモリ使用率「４５％」を示す。

[0125] マネージャノード２０１は、エンジンノード１０１から定期的にＣＰＵ使用率およびメモリ使用率を受信し、エンジンノードテーブル１７００を更新する。マネージャノード２０１は、エンジンノードテーブル１７００に基づいて、エンジンノード１０１の負荷を判断し、負荷が規定値より高くなったエンジンノード１０１のデータ群を他のエンジンノード１０１に割り当てる。

[0126] （クエリテーブル１８００の一例）

図１８は、クエリテーブル１８００の一例を示す説明図である。図１８において、クエリテーブル１８００は、クエリＩＤ、エンジンノードＩＤ、クエリ透過率、部分復元、および並列可能のフィールドを有し、各フィールドに情報を設定することで、クエリ情報（例えば、クエリ情報１８００－１～１８００－５）をレコードとして記憶する。

[0127] ここで、クエリＩＤは、クエリの識別子である。エンジンノードＩＤは、クエリＩＤのクエリが実行されるエンジンノード１０１の識別子である。クエリ透過率は、エンジンノード１０１でのデータ１１１が破棄される割合である。部分復元は、クエリＩＤのクエリで所定の部分１１２のデシリアライズが有効であるか無効であるかを示す情報である。並列可能は、クエリＩＤのクエリが並列クエリであるか否か示す情報である。並列クエリとは、複数のノードで分散して実行可能なクエリである。例えば、クエリ情報１８００－１は、クエリＩＤ「Ｑ１」のクエリは、エンジンノードＩＤ「１」のエンジンノード１０１で実行され、クエリ透過率「５％」、部分復元「有効」および並列可能「可能」を示す。

[0128] マネージャノード２０１は、クエリ透過率を、クエリを実行するエンジンノード１０１から取得し、所定の部分１１２のデシリアライズが有効である

か否かをコンパイラノード２０２から取得する。マネージャノード２０１は、クエリテーブル１８００に基づいて、Ｖｎｏｄｅを移動する際、クエリが並列可能であるか確認し、Ｖｎｏｄｅを割り当てるエンジンノード１０１を決定する。

[0129] （Ｖｎｏｄｅ情報テーブル１９００の一例）

図１９は、Ｖｎｏｄｅ情報テーブル１９００の一例を示す説明図である。図１９において、Ｖｎｏｄｅ情報テーブル１９００は、ＶｎｏｄｅＩＤ、ｉｄ、クエリＩＤ、およびクエリ透過率のフィールドを有し、各フィールドに情報を設定することで、Ｖｎｏｄｅ情報（例えば、Ｖｎｏｄｅ情報１９００－１～１９００－４）をレコードとして記憶する。

[0130] ここで、ＶｎｏｄｅＩＤは、Ｖｎｏｄｅの識別子である。ｉｄは、データの識別子である。クエリＩＤは、Ｖｎｏｄｅで実行されるクエリの識別子である。クエリ透過率は、ＶｎｏｄｅＩＤでの、データ１１１が破棄される割合である。例えば、Ｖｎｏｄｅ情報１９００－１は、Ｖｎｏｄｅ「Ｄ１」が「ｉｄ＝０００～０９９」のデータを含み、クエリＩＤ「Ｑ１」のクエリが実行され、クエリ透過率「７％」を示す。

[0131] マネージャノード２０１は、Ｖｎｏｄｅごとに、Ｖｎｏｄｅで実行するクエリＩＤとクエリ透過率を記憶する。マネージャノード２０１は、Ｖｎｏｄｅごとのクエリ透過率を、クエリを実行するエンジンノード１０１から取得する。マネージャノード２０１は、Ｖｎｏｄｅテーブル１９００に基づいて、移動するＶｎｏｄｅを決定する。

[0132] （ルーティングテーブル２０００の一例）

図２０は、ルーティングテーブル２０００の一例を示す説明図である。図２０において、ルーティングテーブル２０００は、クエリＩＤ、ＶｎｏｄｅＩＤ、およびエンジンノードＩＤのフィールドを有し、各フィールドに情報を設定することで、ルーティング情報（例えばルーティング情報２０００－１～２０００－２）をレコードとして記憶する。

[0133] ここで、クエリＩＤは、クエリの識別子である。ＶｎｏｄｅＩＤは、Ｖｎ

ｏｄｅの識別子である。エンジンノードＩＤは、エンジンノード１０１の識別子である。例えば、ルーティング情報２０００－１は、クエリＩＤ「Ｑ１」、Ｖｎｏｄｅ「Ｄ１」は、エンジンノード１０１のエンジンノードＩＤ「１」に転送されることを示す。

[0134] エンジンノード１０１の送信部５０６は、ルーティングテーブル２０００に基づいて、シリアライズしたデータ１１１を送信するエンジンノード１０１を決定する。

[0135] （エンジンノード１０１の機能的構成例）

実施の形態３にかかるエンジンノード１０１の機能的構成例は、実施の形態２と同じであるため、図示を省略する。また、実施の形態１および実施の形態２で説明した箇所と同一箇所については、説明を省略する。

[0136] 受信部５０１は、シリアライズしたデータ１１１を受信したこと、およびデータ１１１を含むＶｎｏｄｅを識別する情報を制御部１００１に通知する機能をさらに有する。データ１１１を含むＶｎｏｄｅを識別する情報は、ＶｎｏｄｅＩＤである。

[0137] 選別部５０３は、データ１１１を破棄しないこと、およびデータ１１１を含むＶｎｏｄｅを識別する情報を制御部１００１に通知する機能をさらに有する。

[0138] 制御部１００１は、データ１１１を破棄しないと判断した回数とデータ１１１を受信した回数から、Ｖｎｏｄｅごとのクエリ透過率を計算し、マネージャノード２０１に通知する機能をさらに有する。

[0139] （マネージャノード２０１の機能的構成例）

実施の形態３にかかるマネージャノード２０１の機能的構成例は、実施の形態２と同じであるため、図示を省略する。また、実施の形態１および実施の形態２で説明した箇所と同一箇所については、説明を省略する。

[0140] 受信部８０１は、エンジンノード１０１からＶｎｏｄｅごとのクエリ透過率を受信する機能を有する。受信部８０１は、受信したＶｎｏｄｅごとのクエリ透過率を管理部８０２に渡す。

[0141] 管理部８０２は、Ｖｎｏｄｅごとのクエリ透過率でクエリテーブル１８００を更新し、Ｖｎｏｄｅを他のエンジンノード１０１に移動する機能を有する。

[0142] （エンジンノード１０１の透過率処理）

図２１は、実施の形態３にかかるエンジンノード１０１の透過率処理手順の一例を示すフローチャートである。図２１のフローチャートにおいて、まず、制御部１００１は、データ１１１を破棄しないと判断した回数とデータ１１１を受信した回数からエンジンノード１０１全体のクエリ透過率を計算する（ステップＳ２１０１）。次に、制御部１００１は、データ１１１を破棄しないと判断した回数とデータ１１１を受信した回数から、Ｖｎｏｄｅごとのクエリ透過率を計算する（ステップＳ２１０２）。

[0143] 制御部１００１は、計算したエンジンノード１０１全体のクエリ透過率とＶｎｏｄｅごとのクエリ透過率をマネージャノード２０１に通知する（ステップＳ２１０３）。これにより、本フローチャートによる一連の処理は終了する。本フローチャートを実行することにより、エンジンノード１０１全体のクエリ透過率とＶｎｏｄｅごとのクエリ透過率がマネージャノード２０１に通知される。

[0144] （マネージャノード２０１のＶｎｏｄｅ移動処理）

図２２は、実施の形態３にかかるマネージャノード２０１のＶｎｏｄｅ移動処理手順の一例を示すフローチャートである。図２２のフローチャートにおいて、まず、管理部８０２は、エンジンノードテーブル１７００から、ＣＰＵ使用率またはメモリ使用率が規定値より高くなったエンジンノード１０１が存在するか否かを確認する（ステップＳ２２０１）。ＣＰＵ使用率またはメモリ使用率が規定値より高くなったエンジンノード１０１が存在する場合（ステップＳ２２０１：Ｙｅｓ）、管理部８０２は、エンジンノード１０１で実行しているクエリが並列クエリであるか否かを確認する（ステップＳ２２０２）。管理部８０２は、クエリテーブル１８００を参照して、クエリテーブル１８００から確認する。

[0145] 並列クエリである場合（ステップＳ２２０２：Ｙｅｓ）、管理部８０２は、Ｖｎｏｄｅの移動先のエンジンノード１０１を選択する（ステップＳ２２０３）。例えば、管理部８０２は、エンジンノードテーブル１７００から、ＣＰＵ使用率またはメモリ使用率が規定値より低いエンジンノード１０１を選別する。

[0146] 管理部８０２は、選択したエンジンノード１０１で、所定の部分１１２のデシリアライズが有効であるか無効であるかを確認する（ステップＳ２２０４）。管理部８０２は、例えば、クエリテーブル１８００のクエリ透過率で、所定の部分１１２のデシリアライズが有効であるか無効であるかを確認する。所定の部分１１２のデシリアライズが有効である場合（ステップＳ２２０４：Ｙｅｓ）、管理部８０２は、クエリ透過率が所定値$\beta$以下のＶｎｏｄｅを選択したエンジンノード１０１に割り当てる（ステップＳ２２０５）。

[0147] 所定の部分１１２のデシリアライズが無効である場合（ステップＳ２２０４：Ｎｏ）、管理部８０２は、クエリ透過率が所定値$\gamma$以上のＶｎｏｄｅを選択したエンジンノード１０１に割り当てる（ステップＳ２２０６）。管理部８０２は、例えば、データ群テーブル１９０２から、Ｖｎｏｄｅごとの透過率を参照し、所定値$\beta$以下のＶｎｏｄｅおよび所定値$\gamma$以上のＶｎｏｄｅを確認する。

[0148] また、ＣＰＵ使用率またはメモリ使用率が規定値より高くなったエンジンノード１０１が存在しない場合（ステップＳ２２０１：Ｎｏ）、および並列クエリでない場合（ステップＳ２２０２：Ｎｏ）、管理部８０２は、処理を終了する。これにより、本フローチャートによる一連の処理は終了する。本フローチャートを実行することにより、Ｖｎｏｄｅを適切なエンジンノード１０１に割り当てることができる。

[0149] 以上説明したように、エンジンノード１０１は、データ１１１を受信した回数に対する、データ１１１を破棄しないと判断した回数の第１の割合を算出する。また、エンジンノード１０１は、Ｖｎｏｄｅごとに、Ｖｎｏｄｅに含まれるデータ１１１を受信した回数に対する、Ｖｎｏｄｅに含まれるデー

タ１１１を破棄しないと判断した回数の第２の割合を算出する。マネージャノード２０１は、複数のエンジンノード１０１の各々が算出した第１の割合と第２の割合とに基づいて、複数のエンジンノード１０１に割り当てるＶｎｏｄｅを決定する。例えば、マネージャノード２０１は、第１の割合が所定値α以下となるエンジンノード１０１に割り当てるＶｎｏｄｅを、第２の割合が所定値β以下となるＶｎｏｄｅに決定する。また、マネージャノード２０１は、第１の割合が所定値αより大きいエンジンノード１０１に割り当てるＶｎｏｄｅを、第２の割合が所定値γ以上となるＶｎｏｄｅに決定する。

[0150] これにより、マネージャノード２０１は、クエリ透過率が所定値β以下のＶｎｏｄｅを、所定の部分１１２のデシリアライズが有効であるエンジンノード１０１に割り当てることができる。また、マネージャノード２０１は、クエリ透過率が所定値γ以上のＶｎｏｄｅを、所定の部分１１２のデシリアライズが無効であるエンジンノード１０１に割り当てることができる。このため、マネージャノード２０１は、エンジンノード１０１での不要なデシリアライズの削減を考慮した負荷均衡化を行うことができる。

[0151] なお、本実施の形態で説明した処理プログラムは、予め用意されたプログラムをパーソナル・コンピュータやワークステーション等のコンピュータで実行することにより実現することができる。本処理プログラムは、ハードディスク、フレキシブルディスク、ＣＤ－ＲＯＭ、ＭＯ、ＤＶＤ等のコンピュータで読み取り可能な記録媒体に記録され、コンピュータによって記録媒体から読み出されることによって実行される。また、本処理プログラムは、インターネット等のネットワークを介して配布してもよい。

## 符号の説明

[0152] １０１　エンジンノード

２０１　マネージャノード

２０２　コンパイラノード

５０１　受信部

５０２　復元部

５０３　選別部

５０４　処理部

５０５　変換部

５０６　送信部

８０１　受信部

８０２　管理部

８０３　通知部

９０１　構文解析部

９０２　復元コスト検証部

９０３　分配部

１００１　制御部

# 請求の範囲

[請求項1] コンピュータに、

データを受信し、

クエリの処理対象となるデータを選別するために参照される部分を特定する情報に基づいて、受信した前記データの所定の部分を復元し、

復元した前記データの所定の部分と前記処理対象となるデータを選別するための条件とに基づいて、受信した前記データを破棄するか否かを判断し、

破棄しないと判断した場合に、受信した前記データを復元する、

処理を実行させることを特徴とする処理プログラム。

[請求項2] 前記所定の部分を復元する処理は、

前記処理対象となるデータを選別するために参照される部分に含まれるパラメータが、所定の数以下であり、かつ、所定の型である場合に、受信した前記データの所定の部分を復元することを特徴とする請求項１に記載の処理プログラム。

[請求項3] 前記所定の部分を復元する処理は、

前記処理対象となるデータを選別するために参照される部分を復元するコストが、前記処理対象となるデータを復元するコスト未満である場合に、受信した前記データの所定の部分を復元することを特徴とする請求項１または２に記載の処理プログラム。

[請求項4] 前記データを復元する処理は、

受信した前記データを破棄しないと判断し、かつ、前記クエリを実行した後に実行される他のクエリに参照される情報が前記データに含まれる場合は、当該データを復元しないことを特徴とする請求項１～３のいずれか一つに記載の処理プログラム。

[請求項5] 前記コンピュータに、

前記データを受信した回数に対する、受信した前記データを破棄し

ないと判断した回数の割合を算出する処理を実行させ、

前記所定の部分を復元する処理は、

算出した前記割合が所定値以下である場合、受信した前記データの所定の部分を復元することを特徴とする請求項１～４のいずれか一つに記載の処理プログラム。

[請求項6] コンピュータが、

データを受信し、

クエリの処理対象となるデータを選別するために参照される部分を特定する情報に基づいて、受信した前記データの所定の部分を復元し、

復元した前記データの所定の部分と前記処理対象となるデータを選別するための条件とに基づいて、受信した前記データを破棄するか否かを判断し、

破棄しないと判断した場合に、受信した前記データを復元する、

処理を実行することを特徴とする処理方法。

[請求項7] データを受信したことに応じて、クエリの処理対象となるデータを選別するために参照される部分を特定する情報に基づいて、受信した前記データの所定の部分を復元し、復元した前記データの所定の部分と前記処理対象となるデータを選別するための条件とに基づいて、前記データを破棄するか否かを判断し、破棄しないと判断した場合に、前記データを復元する情報処理装置を有することを特徴とする処理システム。

[請求項8] 前記処理対象となるデータを選別するために参照される部分に含まれるパラメータが、所定の数以下であり、かつ、所定の型であるか否かを判定する制御装置をさらに有し、

前記情報処理装置は、

前記パラメータが、所定の数以下であり、かつ、所定の型であると前記制御装置が判定した場合に、前記データの所定の部分を復元する

ことを特徴とする請求項７に記載の処理システム。

[請求項9] 前記処理対象となるデータを選別するために参照される部分を復元するコストが、前記処理対象となるデータを復元するコスト未満であるか否かを判定する制御装置をさらに有し、

前記情報処理装置は、

前記処理対象となるデータを復元するコスト未満であると前記制御装置が判定した場合に、前記データの所定の部分を復元することを特徴とする請求項７または８に記載の処理システム。

[請求項10] 前記クエリを実行した後に実行される他のクエリに参照される情報が前記データに含まれるか否かを判定する制御装置をさらに有し、

前記情報処理装置は、

前記データを破棄しないと判断し、かつ、前記他のクエリに参照される情報が前記データに含まれると前記制御装置が判定した場合は、当該データを復元しないことを特徴とする請求項７～９のいずれか一つに記載の処理システム。

[請求項11] 前記情報処理装置が前記データを受信した回数に対する、前記情報処理装置が前記データを破棄しないと判断した回数の第１の割合が、第１の閾値以下であるか否かを判定する制御装置をさらに有し、

前記情報処理装置は、

前記第１の割合を算出し、前記第１の割合が前記第１の閾値以下であると前記制御装置が判定した場合に、前記データの所定の部分を復元することを特徴とする請求項７～１０のいずれか一つに記載の処理システム。

[請求項12] 前記情報処理装置は複数存在し、

前記情報処理装置は、

複数のデータから分割されたいずれかのデータ群に含まれるデータを受信したことに応じて、前記データ群に含まれるデータを受信した回数に対する、前記データ群に含まれるデータを破棄しないと判断し

た回数の第２の割合を算出し、

前記制御装置は、

複数の前記情報処理装置の各々が算出した前記第１の割合と前記第２の割合とに基づいて、複数の前記情報処理装置に割り当てる前記データ群を決定することを特徴とする請求項１１に記載の処理システム。

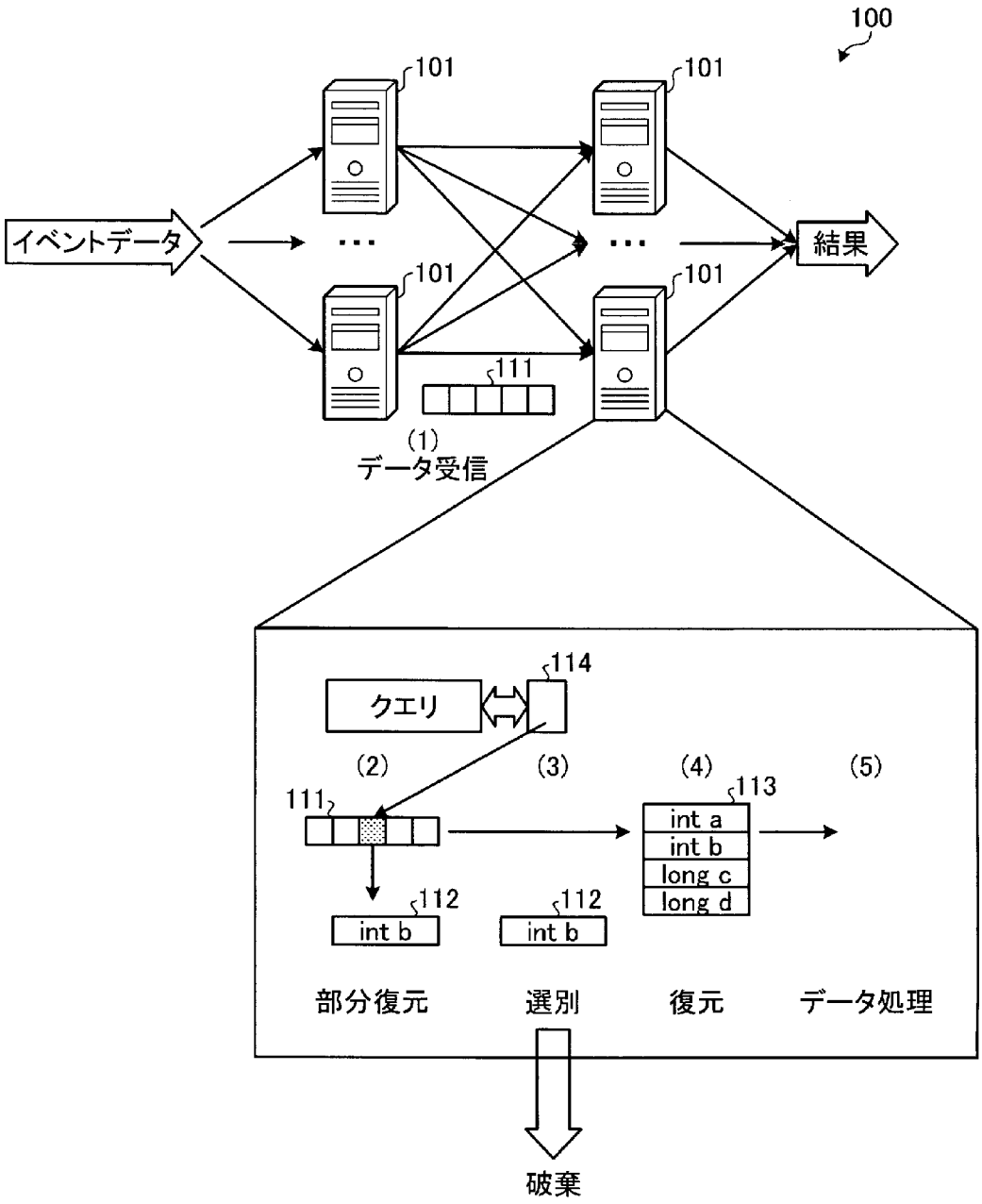
［請求項13］ 前記制御装置は、

前記第１の割合が前記第１の閾値以下となる情報処理装置に割り当てる前記データ群を、前記第２の割合が第２の閾値以下となるデータ群に決定することを特徴とする請求項１２に記載の処理システム。

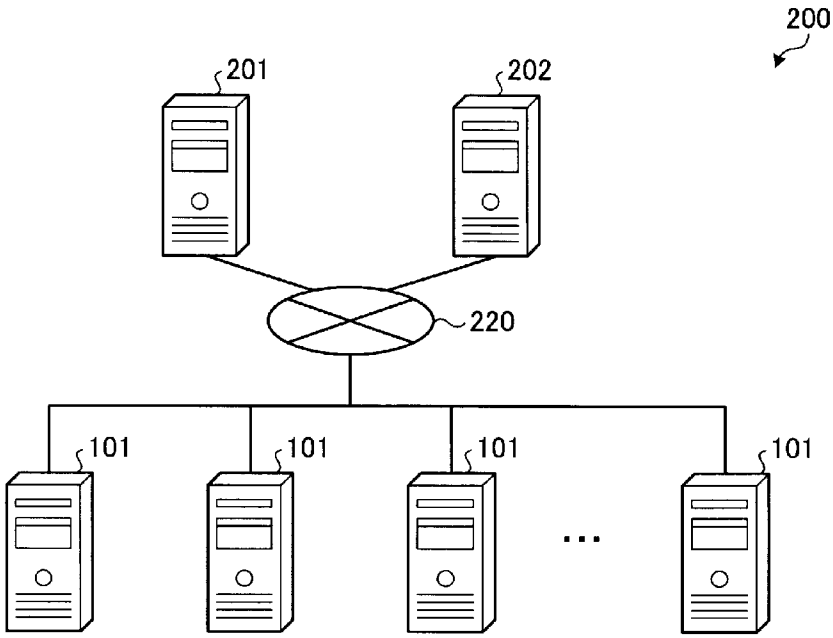
［請求項14］ 前記制御装置は、

前記第１の割合が前記第１の閾値より大きい情報処理装置に割り当てる前記データ群を、前記第２の割合が第３の閾値以上となるデータ群に決定することを特徴とする請求項１２または１３に記載の処理システム。

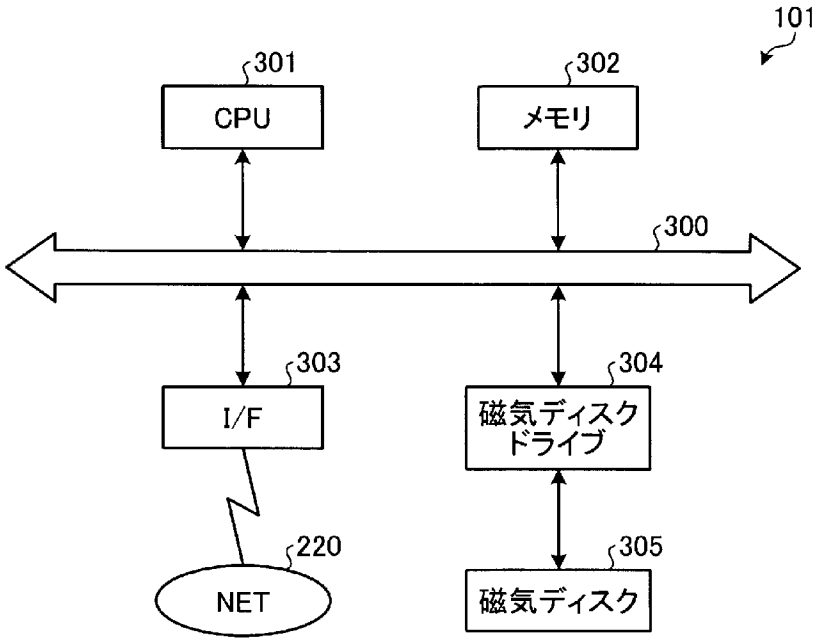
[図1]


100
101
101
101
101
イベントデータ
結果
111
(1)
データ受信
114
クエリ
(2)
(3)
(4)
(5)
111
113
int a
int b
long c
long d
112
112
int b
int b
部分復元
選別
復元
データ処理
破棄

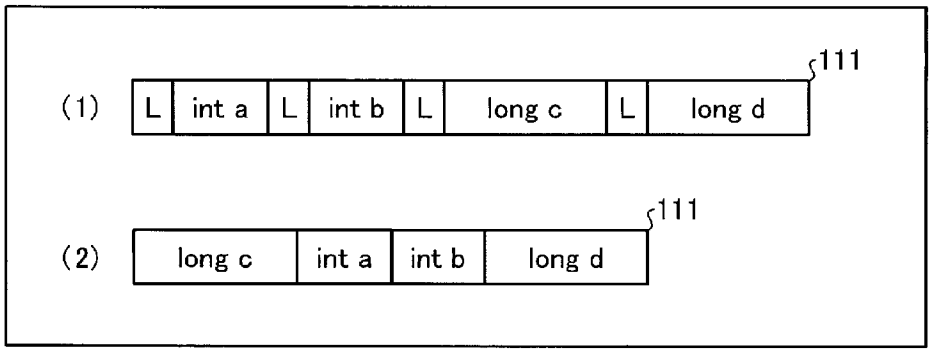
[図2]


200
201
202
220
101
101
101
101

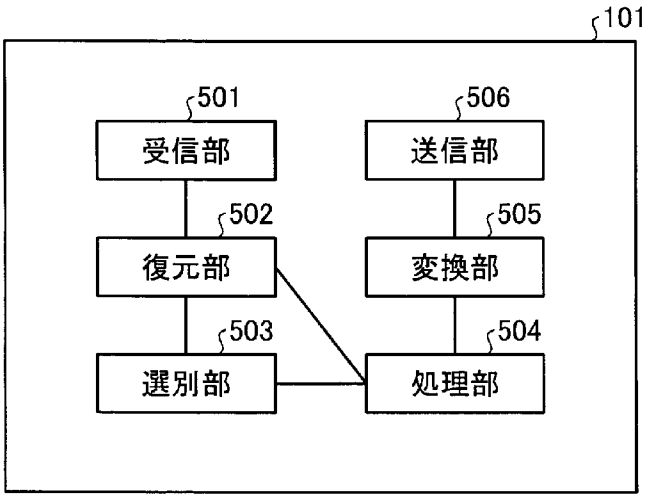
[図3]


101
301
CPU
302
メモリ
300
303
I/F
304
磁気ディスク
ドライブ
220
NET
305
磁気ディスク

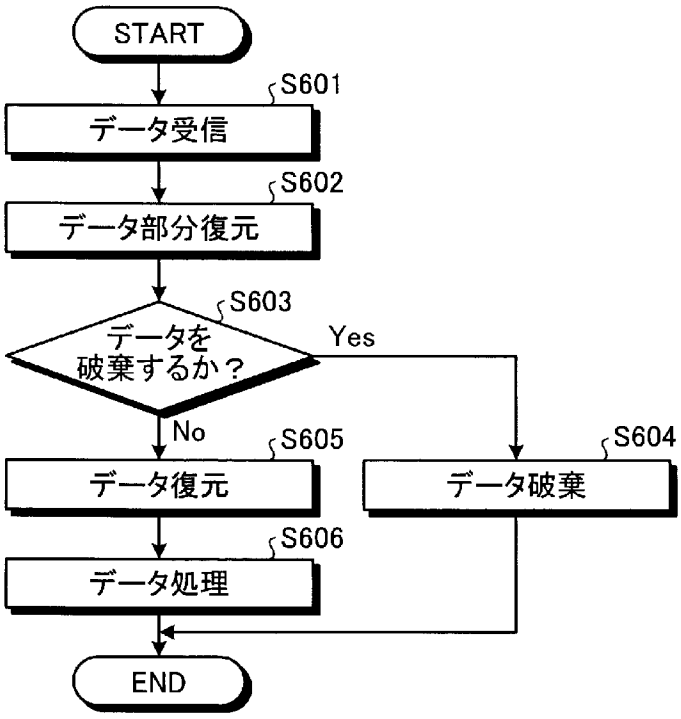
[図4]


111
(1)
L
int a
L
int b
L
long c
L
long d
111
(2)
long c
int a
int b
long d

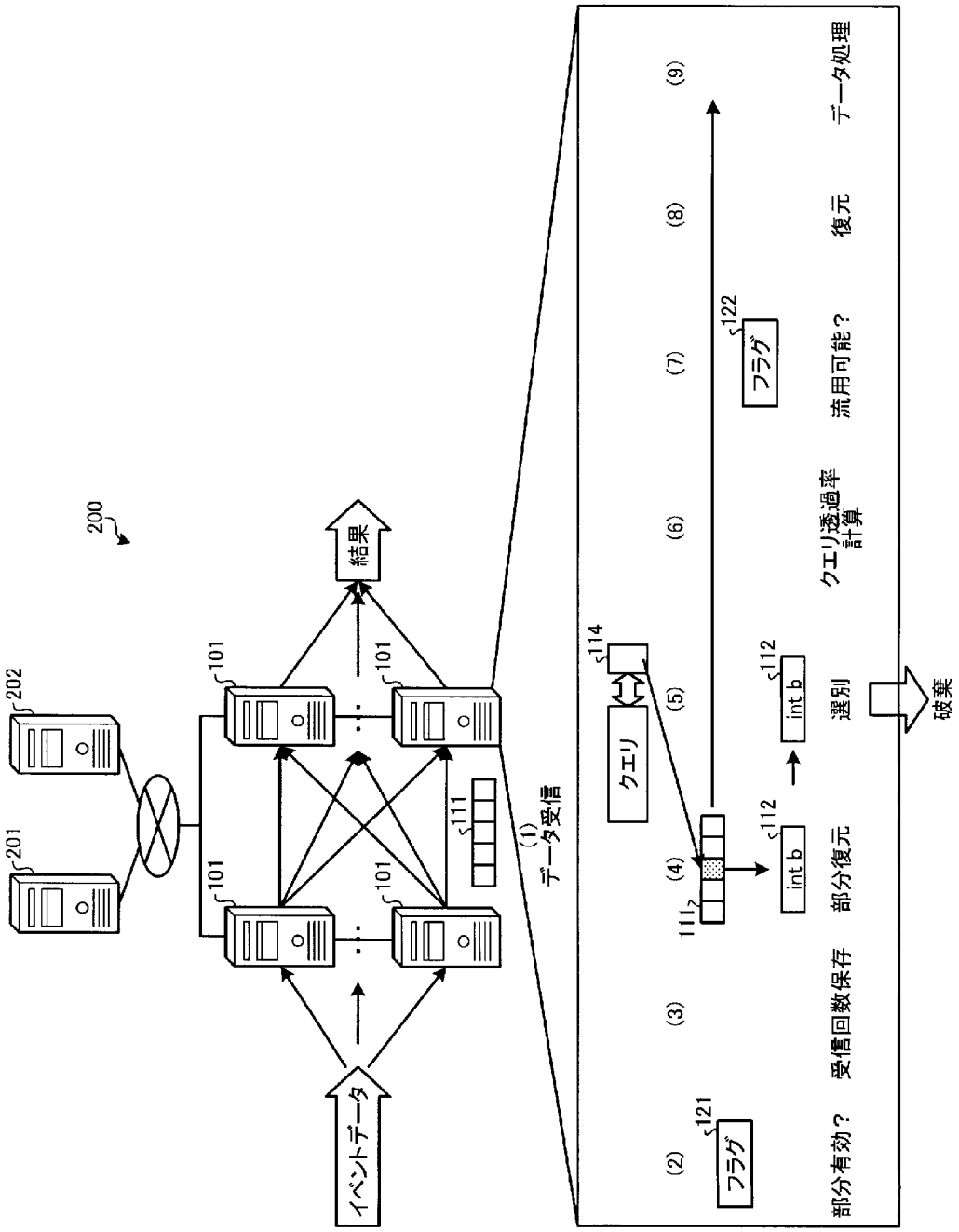
[図5]


101
501
受信部
506
送信部
502
復元部
505
変換部
503
選別部
504
処理部

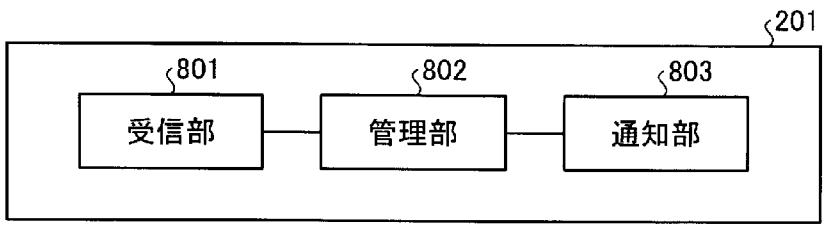
[図6]


START
S601
データ受信
S602
データ部分復元
S603
データを
破棄するか？
Yes
No
S605
データ復元
S604
データ破棄
S606
データ処理
END

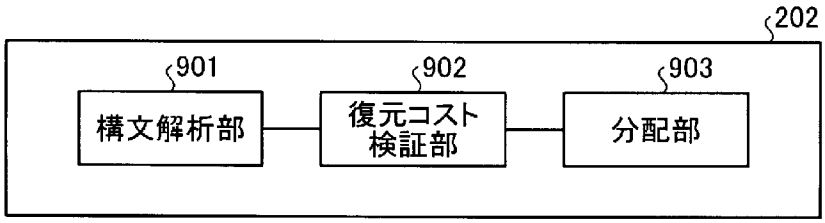
[図7]


200
結果
101
202
201
111
イベントデータ
(1)
データ受信
114
クエリ
(5)
112
int b
選別
破棄
(9)
データ処理
(8)
復元
(7)
122
フラグ
流用可能？
(6)
クエリ透過率
計算
(4)
112
int b
部分復元
111
(3)
受信回数保存
(2)
121
フラグ
部分有効？

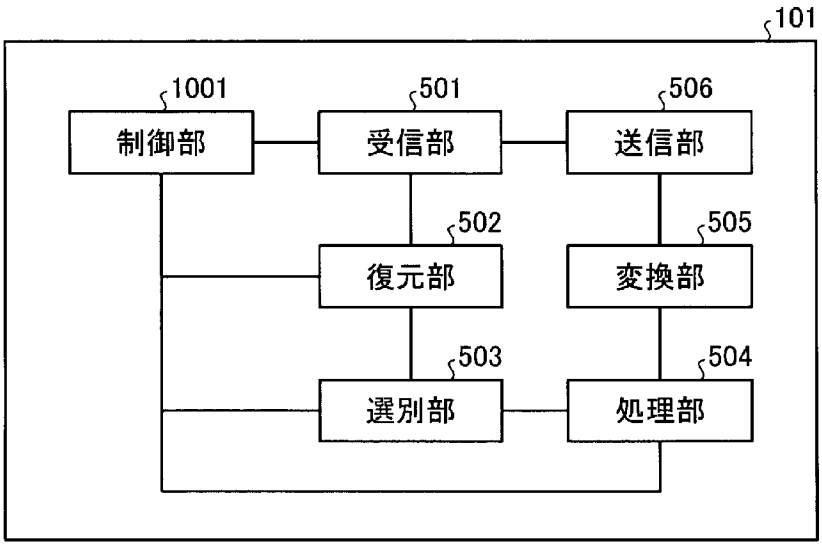
[図8]


201
801
受信部
802
管理部
803
通知部

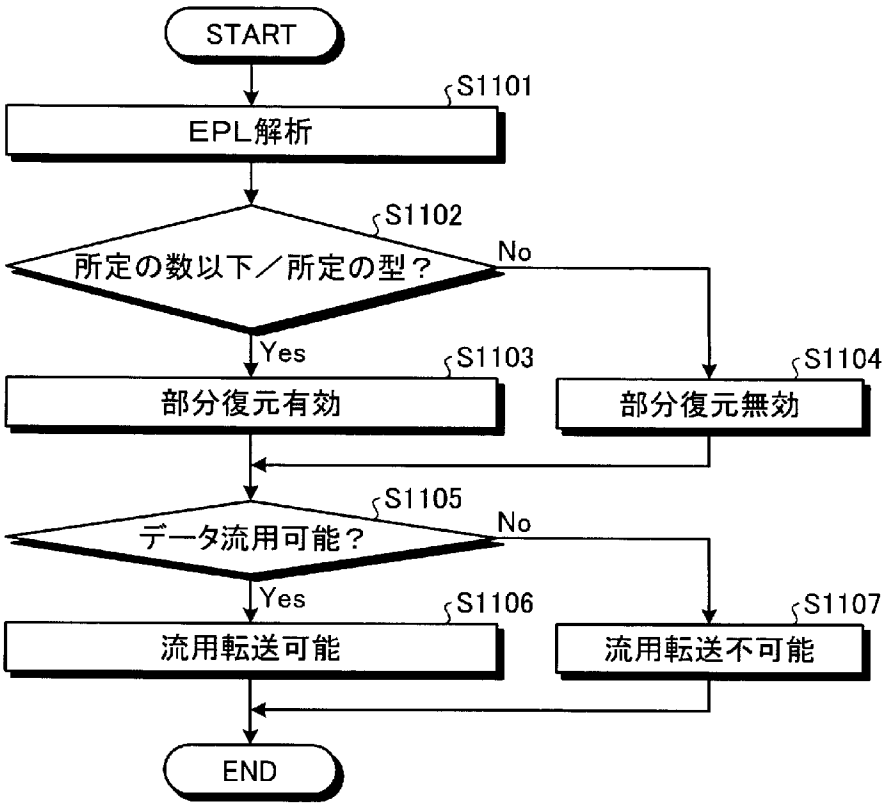
[図9]


202
901
構文解析部
902
復元コスト検証部
903
分配部

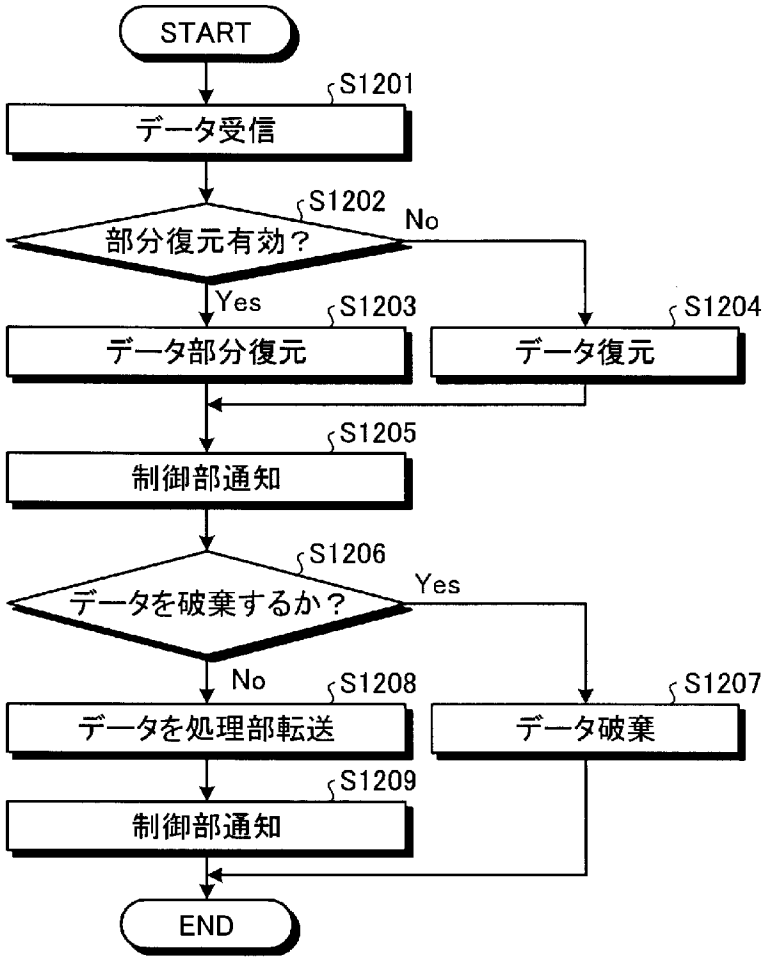
[図10]


101
1001
制御部
501
受信部
506
送信部
502
復元部
505
変換部
503
選別部
504
処理部

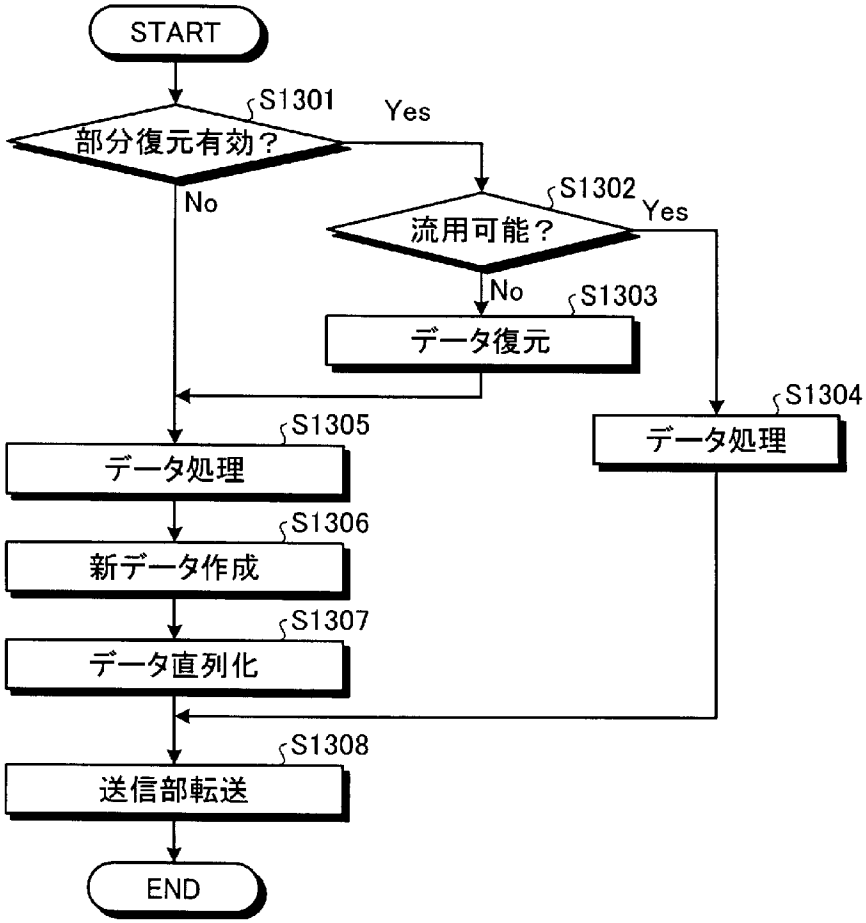
[図11]


START
S1101
EPL解析
S1102
所定の数以下／所定の型？
No
Yes
S1103
部分復元有効
S1104
部分復元無効
S1105
データ流用可能？
No
Yes
S1106
流用転送可能
S1107
流用転送不可能
END

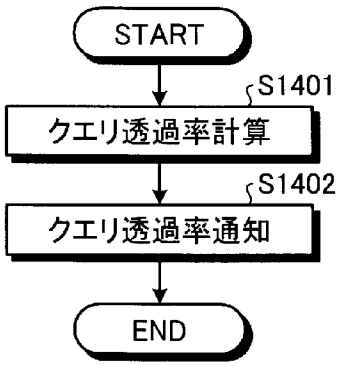
[図12]


START
S1201
データ受信
S1202
部分復元有効？
No
Yes
S1203
データ部分復元
S1204
データ復元
S1205
制御部通知
S1206
データを破棄するか？
Yes
No
S1208
データを処理部転送
S1207
データ破棄
S1209
制御部通知
END

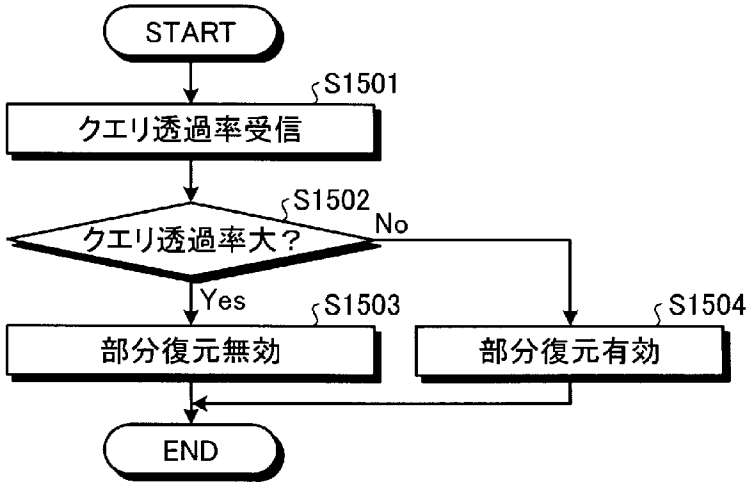
[図13]


START
S1301
部分復元有効？
Yes
No
S1302
流用可能？
Yes
No
S1303
データ復元
S1304
データ処理
S1305
データ処理
S1306
新データ作成
S1307
データ直列化
S1308
送信部転送
END

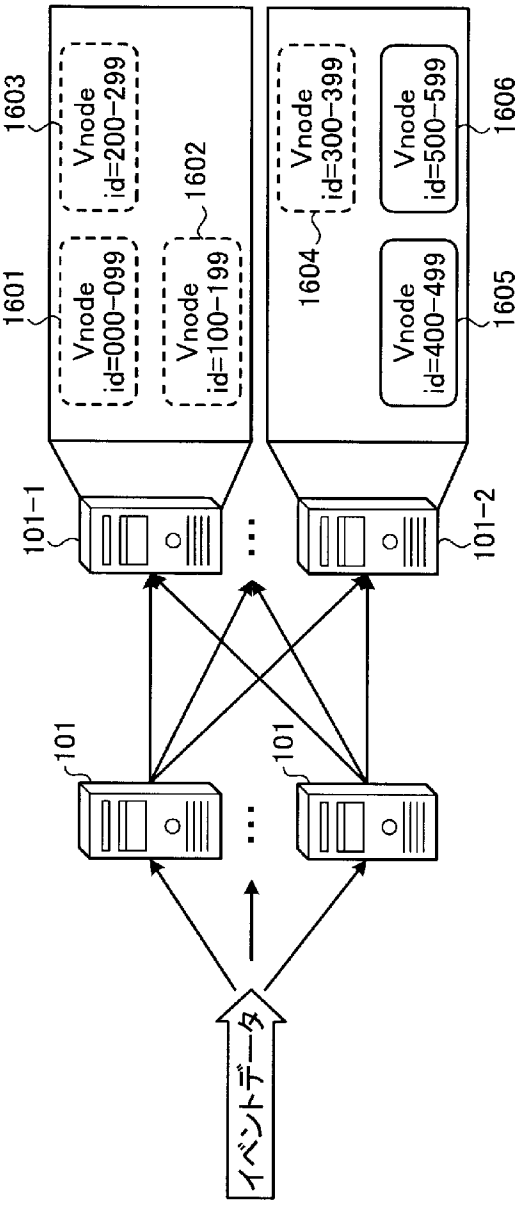
[図14]


START
S1401
クエリ透過率計算
S1402
クエリ透過率通知
END

[図15]


START
S1501
クエリ透過率受信
S1502
クエリ透過率大？
No
Yes
S1503
部分復元無効
S1504
部分復元有効
END

[図16]


1603
Vnode id=200-299
1602
1601
Vnode id=000-099
Vnode id=100-199
Vnode id=300-399
1604
Vnode id=500-599
1606
Vnode id=400-499
1605
101-1
101-2
101
101
イベントデータ

[図17]

1700

| | エンジンノードID | CPU使用率 | メモリ使用率 |
|---|---|---|---|
| 1700-1 | 1 | 62% | 45% |
| 1700-2 | 2 | 24% | 43% |
| 1700-3 | 3 | 13% | 15% |
| | ... | ... | ... |

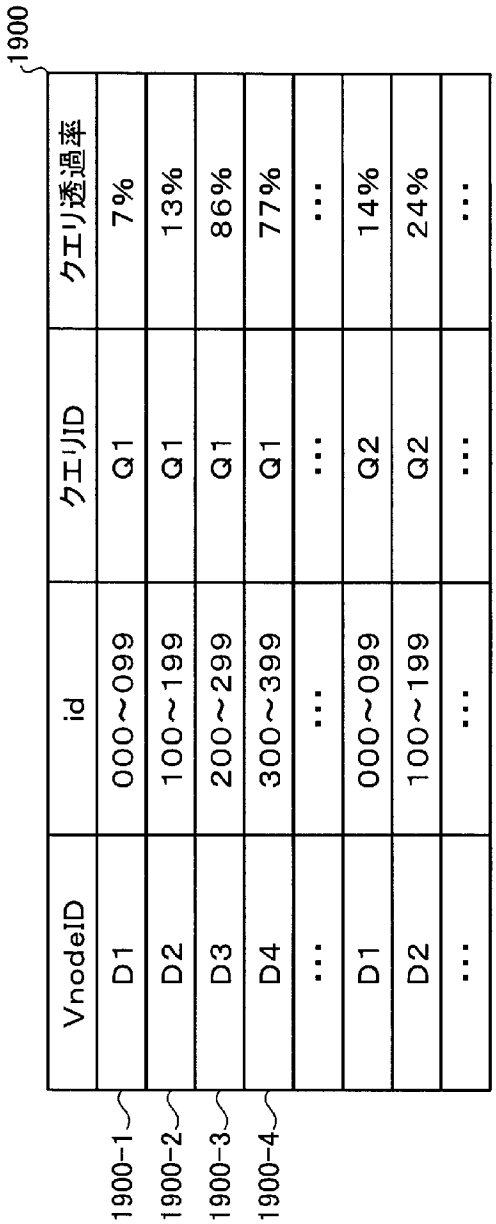
[図18]

1800

| クエリID | エンジンノードID | クエリ透過率 | 部分復元 | 並列可能 |
|---|---|---|---|---|
| Q1 | 1 | 5% | 有効 | 可能 |
| Q1 | 2 | 3% | 有効 | 可能 |
| Q1 | 3 | 66% | 無効 | 可能 |
| Q2 | 2 | 25% | 有効 | 可能 |
| Q4 | 3 | 12% | 有効 | 不可能 |
| ... | ... | ... | ... | ... |

1800-1 1800-2 1800-3 1800-4 1800-5

[図19]

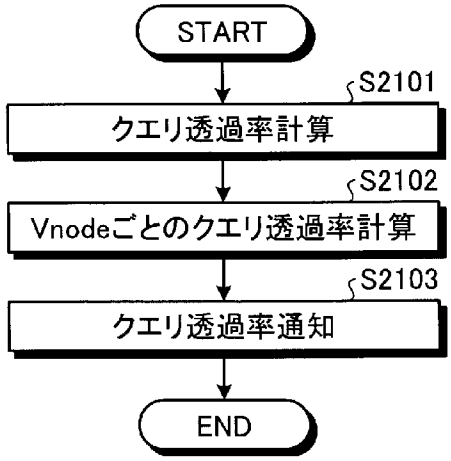



[図20]

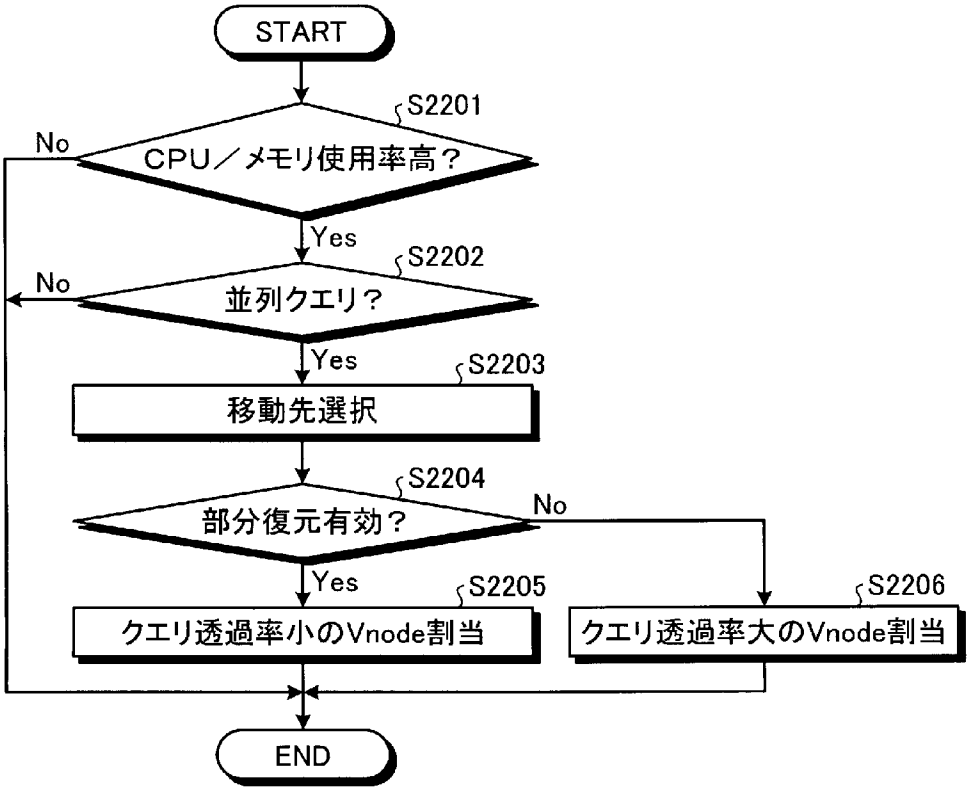
2000

| | クエリID | VnodeID | エンジンノードID |
|---|---|---|---|
| 2000-1 | Q1 | D1 | 1 |
| 2000-2 | Q1 | D2 | 2 |
| | ... | ... | ... |
| | Q2 | D1 | 3 |
| | Q2 | D2 | 3 |
| | ... | ... | ... |

[図21]


START
S2101
クエリ透過率計算
S2102
Vnodeごとのクエリ透過率計算
S2103
クエリ透過率通知
END

[図22]


START
S2201
CPU／メモリ使用率高？
No
Yes
S2202
並列クエリ？
No
Yes
S2203
移動先選択
S2204
部分復元有効？
No
Yes
S2205
クエリ透過率小のVnode割当
S2206
クエリ透過率大のVnode割当
END

# INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2013/070664

A. CLASSIFICATION OF SUBJECT MATTER
*G06F17/30*(2006.01)i, *G06F9/50*(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)
G06F17/30, G06F9/50, G06F12/00, G06F13/00

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 | 1996-2013 |
| Kokai Jitsuyo Shinan Koho | 1971-2013 | Toroku Jitsuyo Shinan Koho | 1994-2013 |

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 2011-8614 A (NEC Corp.), 13 January 2011 (13.01.2011), entire text; all drawings (Family: none) | 1-14 |
| A | JP 2013-114626 A (Fujitsu Ltd.), 10 June 2013 (10.06.2013), entire text; all drawings & US 2013/0138686 A1 & EP 2600259 A2 & CN 103136322 A | 1-14 |
| A | WO 2011/158478 A1 (NEC Corp.), 22 December 2011 (22.12.2011), entire text; all drawings & US 2013/0091203 A1 | 1-14 |

☐ 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 | Date of mailing of the international search report |
|---|---|
| 18 September, 2013 (18.09.13) | 01 October, 2013 (01.10.13) |
| Name and mailing address of the ISA/ Japanese Patent Office<br>Facsimile No. | Authorized officer<br>Telephone No. |

国際調査報告

国際出願番号　ＰＣＴ／ＪＰ２０１３／０７０６６４

Ａ．　発明の属する分野の分類（国際特許分類（ＩＰＣ））

Int.Cl.　G06F17/30(2006.01)i, G06F9/50(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（ＩＰＣ））

Int.Cl.　G06F17/30, G06F9/50, G06F12/00, G06F13/00

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 | １９２２－１９９６年 |
| 日本国公開実用新案公報 | １９７１－２０１３年 |
| 日本国実用新案登録公報 | １９９６－２０１３年 |
| 日本国登録実用新案公報 | １９９４－２０１３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 2011-8614 A（日本電気株式会社）2011.01.13, 全文, 全図 (ファミリーなし) | １－１４ |
| Ａ | JP 2013-114626 A（富士通株式会社）2013.06.10, 全文, 全図 & US 2013/0138686 A1 & EP 2600259 A2 & CN 103136322 A | １－１４ |
| Ａ | WO 2011/158478 A1（日本電気株式会社）2011.12.22, 全文,全図 & US 2013/0091203 A1 | １－１４ |

☐　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 | 国際調査報告の発送日 |
|---|---|
| １８．０９．２０１３ | ０１．１０．２０１３ |


国際調査機関の名称及びあて先
日本国特許庁（ＩＳＡ／ＪＰ）
郵便番号１００－８９１５
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）　５Ｍ　３６５９
吉田　誠
電話番号　０３－３５８１－１１０１　内線　３５９９