

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第6260359号
(P6260359)

(45) 発行日 平成30年1月17日(2018.1.17)

(24) 登録日 平成29年12月22日(2017.12.22)

(51) Int.Cl.

F I

G 0 6 F 12/00 (2006.01)

G 0 6 F 12/00 5 1 1 C

G 0 6 F 17/30 (2006.01)

G 0 6 F 17/30 4 1 5

請求項の数 5 (全 24 頁)

(21) 出願番号 特願2014-45042 (P2014-45042)
 (22) 出願日 平成26年3月7日(2014.3.7)
 (65) 公開番号 特開2015-170170 (P2015-170170A)
 (43) 公開日 平成27年9月28日(2015.9.28)
 審査請求日 平成28年11月2日(2016.11.2)

(73) 特許権者 000005223
 富士通株式会社
 神奈川県川崎市中原区上小田中4丁目1番
 1号
 (74) 代理人 100092978
 弁理士 真田 有
 (74) 代理人 100112678
 弁理士 山本 雅久
 (72) 発明者 立岩 恭也
 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
 審査官 桜井 茂行

最終頁に続く

(54) 【発明の名称】 データ分割処理プログラム、データ分割処理装置及びデータ分割処理方法

(57) 【特許請求の範囲】

【請求項1】

コンピュータに、

処理対象である、バイナリデータが複数に分割された部分バイナリデータのデータ区切り位置を、前記バイナリデータを構成する所定のデータの特性に基づき検出し、

検出したデータ区切り位置を示す情報を記憶手段に登録し、

前記記憶手段に登録された他の部分バイナリデータに関するデータ区切り位置を示す情報に基づき、前記部分バイナリデータにおいて検出したデータ区切り位置が正当であることを判定し、

前記処理対象の部分バイナリデータについて検出したデータ区切り位置と、前記記憶手段に登録された、前記バイナリデータにおいて前記処理対象の部分バイナリデータと隣接する部分バイナリデータについてのデータ区切り位置とが連続する場合に、検出したデータ区切り位置が正当であると判断する

処理を実行させることを特徴とする、データ分割処理プログラム。

【請求項2】

前記部分バイナリデータを構成する個々の単位レコードに含まれるデータ長情報に基づき、当該部分バイナリデータの位置を特定する

処理を前記コンピュータに実行させることを特徴とする、請求項1記載のデータ分割処理プログラム。

【請求項3】

10

20

正当であると判断された前記データ区切り位置に基づき、前記バイナリデータのデータ区切り位置を確定する

処理を前記コンピュータに実行させることを特徴とする、請求項 1 又は 2 記載のデータ分割処理プログラム。

【請求項 4】

バイナリデータを複数に分割した部分バイナリデータを処理するデータ分割処理装置であって、

処理対象の部分バイナリデータのデータ区切り位置を、前記バイナリデータを構成する所定のデータの特性に基づき検出する検出部と、

検出したデータ区切り位置を示す情報を記憶手段に登録する登録部と、

前記記憶手段に登録された他の部分バイナリデータに関するデータ区切り位置を示す情報に基づき、前記部分バイナリデータにおいて検出したデータ区切り位置が正当であるかを判定する判定部と、

を備え、

前記判定部は、前記処理対象の部分バイナリデータについて検出したデータ区切り位置と、前記記憶手段に登録された、前記バイナリデータにおいて前記処理対象の部分バイナリデータと隣接する部分バイナリデータについてのデータ区切り位置とが連続する場合に、検出したデータ区切り位置が正当であると判断することを特徴とする、データ分割処理装置。

【請求項 5】

バイナリデータを複数に分割して複数の処理部により処理する分割処理方法であって、

前記複数の処理部のそれぞれに、前記バイナリデータの一部である部分バイナリデータをそれぞれ割り当て、

前記複数の処理部はそれぞれ、割り当てられた部分バイナリデータのデータ区切り位置を、前記バイナリデータを構成する所定のデータの特性に基づき検出するとともに、検出したデータ区切り位置を示す情報を記憶手段に登録し、

前記複数の処理部はそれぞれ、他の処理部により前記記憶手段に登録されたデータ区切り位置を示す情報に基づき、前記部分バイナリデータにおいて検出したデータ区切り位置が正当であるかを判定し、

前記処理対象の部分バイナリデータについて検出したデータ区切り位置と、前記記憶手段に登録された、前記バイナリデータにおいて前記処理対象の部分バイナリデータと隣接する部分バイナリデータについてのデータ区切り位置とが連続する場合に、検出したデータ区切り位置が正当であると判断する

ことを特徴とする、データ分割処理方法。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、データ分割処理プログラム、データ分割処理装置及びデータ分割処理方法に関する。

【背景技術】

【0002】

近年、ウェブデータ等の大量のデータを処理するための処理システムとして、マップ・リデュース（Map-Reduce）型の分散処理システムが知られている。

マップ・リデュース型の分散処理システムにおいては、分散処理システム上のデータをデータブロックと呼ばれる単位に分割し、これらのデータブロックに対してマップ処理及びリデュース処理を順次適用する。

【0003】

このようなマップ・リデュース型の分散処理システムによれば、各データブロックに対する一連の計算処理を複数の計算ノードで同時に分散して実行することが可能となる。

Hadoop（登録商標）は大規模データを効率的に分散処理・管理するための O S S（Open

10

20

30

40

50

Source Software) のフレームワークであり、主に分析処理で活用されている。このHadoopを基幹系のバッチ処理に適用することにより、データを複数のマシンに分散して処理させ、処理時間が課題となっている大規模バッチ処理の高速化が実現できる。

【 0 0 0 4 】

Hadoopにおいては、マスタノードが複数のスレーブノードにタスクをそれぞれ割り当て、スレーブノードは、マスタノードから割り当てられたマップタスク (Mapタスク) を実行する。

Hadoopでは、ファイルを一定のサイズのブロックに分割し、そのブロック毎にマップタスクで処理を行う。

【 0 0 0 5 】

10

図 1 5 はHadoopのマップ - リデュースフレームワークの動作概要を示す図である。図 1 5 に示す例においては、Hadoop分散ファイルシステム (HDFS : Hadoop Distributed File System) によって管理される 1 9 6 M B のファイルが 6 4 M B のデータサイズを有する 3 つのブロックに分割され、それらを 3 つのマップタスクで並列に処理している。マップタスクから出力されたデータは、Shuffle&sortやReduceタスクを経て、ソートされた結果ファイルとして出力され、H D F S に戻される。

【 0 0 0 6 】

このような分散処理システムにおいて、ファイルが分割されてもデータ自体は分割されないことが重要である。例えるならば、マップタスクに渡される「みかん」というデータが、「みか」と「ん」に分かれてはならない。このように一体として扱われるべきデータが分割されてしまうことを泣き別れという場合がある。

20

このため、6 4 M B でファイル分割が行なわれた後、データの区切り位置を調整する必要がある。Hadoopのデフォルトでは、データの区切りとして改行コードを用い、この改行コードの位置を区切りとして処理することで、データの泣き別れを防いでいる。この区切りの調整処理はカスタマイズ可能であり、例えば任意の文字を区切りとするような、カスタマイズが可能になっている。

【 0 0 0 7 】

図 1 6 はNetCOBOLの可変長レコード順ファイルのフォーマットを例示する図である。

NetCOBOLの可変長レコード順ファイルは、この図 1 6 に示すように、データの前後に 4 バイトのレコード長情報を有する可変長レコードを複数連続することにより構成されている。データは利用者が設定する任意のバイナリ値である。また、データの前後に配置されるレコード長には同一の値が格納される。

30

【 0 0 0 8 】

なお、利用者はレコード長情報を設定・参照する必要はなく、COBOLランタイムシステムが参照・設定を行なう。

図 1 7 はNetCOBOLの可変長レコード順ファイルのレコードイメージを示す図である。

この図 1 7 に示す例においては、1 つの可変長レコード毎に改行して見やすくしているが、実際は複数の可変長レコードが連続している。

【 0 0 0 9 】

NetCOBOLの可変長レコード順ファイルをHadoopで利用する場合、ブロック単位でファイル分割が行われた後、データの区切りを調整することができない。これは、データが利用者が任意に設定可能なバイナリ値であり、改行コードや任意の文字を区切りとする場合に、データ内にこれらの改行コードや区切り文字と同じ情報が含まれるおそれがあり、データの区切り位置を特定できないためである。

40

【 0 0 1 0 】

図 1 8 はNetCOBOLの可変長レコード順ファイルを 2 つに分割する場合のレコードイメージを示す図である。

この図 1 8 に示す例においては、ファイルを 6 4 M B のブロックサイズ単位で分割する場合に、データ長が 1 0 5 バイトのデータが、その途中で分断されてしまうことを示す。

そこで、NetCOBOLでは、レコードの位置を正しく求めてデータ区切り位置を調整するた

50

めに、ブロックサイズによる分断位置から次の可変長レコードの先頭位置までの距離（バイト長）を保持する情報ファイルを予め作成し、この情報ファイルを区切りの調整処理で参照する手法が用いられる。

【 0 0 1 1 】

図 1 9 はNetCOBOLの可変長レコード順ファイルと情報ファイルとを示す図である。

情報ファイルは、入力ファイルである可変長レコード順ファイルを読み込んでレコード長を加算し、ブロックサイズで分断される位置から次のレコード先頭位置までの情報を保持する。

図 1 9 に示す例においては、情報ファイルにおいて、例えばファイルを先頭から 6 4 M B の位置で分割する場合に、その分断位置から次の可変長レコード（データ長が 2 0 バイト）の先頭までのデータ長が 5 5 バイトであることを示す。なお、情報サイズ作成時には、ブロックサイズもパラメータで指定する必要がある。

【先行技術文献】

【特許文献】

【 0 0 1 2 】

【特許文献 1】特開 2 0 1 2 - 1 1 8 6 6 9 号公報

【特許文献 2】特表平 1 0 - 5 0 0 7 9 3 号公報

【特許文献 3】特開平 3 - 6 2 1 3 7 号公報

【発明の概要】

【発明が解決しようとする課題】

【 0 0 1 3 】

しかしながら、このような従来の分散処理システムにおいては、上述した情報ファイルを作成するために、処理対象ファイル全体を、その先頭から可変長レコードのレコード長を順次取り出して加算することで区切り位置を求める。このため、例えば、データサイズが数十～数百 G B であり、レコード件数が数百万件程度の大きなサイズのファイルでは、情報ファイルの作成処理に時間がかかってしまう。例えば 8 0 G B のデータサイズのファイルにおいては、情報ファイルの作成に例えば 1 5 分程度の時間を要する場合もある。

【 0 0 1 4 】

これにより、本来は処理時間の高速化を目的としてHadoopを導入するにもかかわらず、情報ファイルの作成に時間を要し、全体の処理時間で考えるとHadoopによる時間短縮効果が薄れてしまうという課題がある。

1 つの側面では、本発明は、分割処理を高速化できるようにすることを目的とする。

なお、前記目的に限らず、後述する発明を実施するための形態に示す各構成により導かれる作用効果であって、従来の技術によっては得られない作用効果を奏することも本発明の他の目的の 1 つとして位置付けることができる。

【課題を解決するための手段】

【 0 0 1 5 】

このため、このデータ分割処理方法は、バイナリデータを複数に分割して複数の処理部により処理する分割処理方法であって、前記複数の処理部のそれぞれに、前記バイナリデータの一部である部分バイナリデータをそれぞれ割り当て、前記複数の処理部はそれぞれ、割り当てられた部分バイナリデータのデータ区切り位置を、前記バイナリデータを構成する所定のデータの特性に基づき検出するとともに、検出したデータ区切り位置を示す情報を記憶手段に登録し、前記複数の処理部はそれぞれ、他の処理部により前記記憶手段に登録されたデータ区切り位置を示す情報に基づき、前記部分バイナリデータにおいて検出したデータ区切り位置が正当であるかを判定し、前記処理対象の部分バイナリデータについて検出したデータ区切り位置と、前記記憶手段に登録された、前記バイナリデータにおいて前記処理対象の部分バイナリデータと隣接する部分バイナリデータについてのデータ区切り位置とが連続する場合に、検出したデータ区切り位置が正当であると判断する。

【発明の効果】

【 0 0 1 6 】

一実施形態によれば、分割処理を高速化することができる。

【図面の簡単な説明】

【0017】

【図1】実施形態の一例としての分散処理システムの機能構成を模式的に示す図である。

【図2】実施形態の一例としての分散処理システムのサーバのハードウェア構成を例示する図である。

【図3】実施形態の一例としての分散処理システムにおけるノード間共有情報を例示する図である。

【図4】実施形態の一例としての分散処理システムにおける特定部による区切り位置の特定方法を説明するための図である。

【図5】実施形態の一例としての分散処理システムにおける分割ファイルの区切り位置の調整方法を説明するフローチャートである。

【図6】実施形態の一例としての分散処理システムにおけるファイル読込部によるファイルの読み込み処理を説明するフローチャートである。

【図7】実施形態の一例としての分散処理システムにおけるファイル先頭記録部による処理を説明するフローチャートである。

【図8】実施形態の一例としての分散処理システムにおける整合性判定部及びファイル終端記録部による処理を説明するフローチャートである。

【図9】実施形態の一例としての分散処理システムにおける解析対象ファイルを例示する図である。

【図10】実施形態の一例としての分散処理システムにおける解析対象ファイルの分割処理を説明するための図である。

【図11】実施形態の一例としての分散処理システムにおける解析対象ファイルの分割処理を説明するための図である。

【図12】実施形態の一例としての分散処理システムにおける解析対象ファイルの分割処理を説明するための図である。

【図13】実施形態の一例としての分散処理システムの特定部が犯しうる区切り位置の誤検知を例示する図である。

【図14】MicroFocusのレコード順ファイルの構成を例示する図である。

【図15】Hadoopのマップ・リデュースフレームワークの動作概要を示す図である。

【図16】NetCOBOLの可変長レコード順ファイルのフォーマットを例示する図である。

【図17】NetCOBOLの可変長レコード順ファイルのレコードイメージを示す図である。

【図18】NetCOBOLの可変長レコード順ファイルを2つに分割する場合のレコードイメージを示す図である。

【図19】NetCOBOLの可変長レコード順ファイルと情報ファイルとを示す図である。

【発明を実施するための形態】

【0018】

以下、図面を参照して本データ分割処理プログラム、データ分割処理装置及びデータ分割処理方法に係る実施の形態を説明する。ただし、以下に示す実施形態はあくまでも例示に過ぎず、実施形態で明示しない種々の変形例や技術の適用を排除する意図はない。すなわち、本実施形態を、その趣旨を逸脱しない範囲で種々変形して実施することができる。又、各図は、図中に示す構成要素のみを備えるという趣旨ではなく、他の機能等を含むことができる。

【0019】

図1は実施形態の一例としての分散処理システム1の機能構成を模式的に示す図、図2はそのサーバのハードウェア構成を例示する図である。

本分散処理システム1は、複数（図1に示す例では4つ）のサーバ（計算ノード、ノード）10-1～10-4をそなえ、これらの複数のサーバ10-1～10-4で分散して処理を行なう。本分散処理システム1は、例えばHadoop（登録商標）を用いて分散処理を実現するマップ・リデュースシステムである。Hadoopはデータ（ファイル）を複数のマシ

10

20

30

40

50

ンに分散して処理させるオープンソースのプラットフォームであり、公知の技術であるので、その詳細な説明は省略する。

【 0 0 2 0 】

また、本実施形態においては、処理対象のファイルが、図 1 6 , 図 1 7 等に例示した NetCOBOL の規格に沿った可変長レコードファイル（以下、単にファイルという）である例について説明する。すなわち、処理対象のファイルはバイナリデータであり、また、処理対象のファイルは、複数の可変長レコード（単位レコード）を連続することにより構成されている。さらに、個々の可変長レコードにおいては、データの前後に当該データのデータ長を示すレコード長情報が配置されているというデータ構造上の特性（規則性）を有する。なお、可変長レコードにおけるデータよりも前側のレコード長情報を前側レコード長情報といい、データよりも後側に位置するレコード長情報を、後側レコード長情報という場合がある。

10

【 0 0 2 1 】

各サーバ 1 0 - 1 ~ 1 0 - 4 はネットワーク 5 0 を介して相互に通信可能に接続されている。ネットワーク 5 0 は、例えば LAN（Local Area Network）等の通信回線である。

サーバ 1 0 - 1 ~ 1 0 - 4 は、それぞれサーバ機能をそなえたコンピュータ（情報処理装置、処理部）である。各サーバ 1 0 - 1 ~ 1 0 - 4 は、同様の構成を備える。以下、サーバを示す符号としては、複数のサーバのうち 1 つを特定する必要があるときには符号 1 0 - 1 ~ 1 0 - 4 を用いるが、任意のサーバを指すときには符号 1 0 を用いる。

【 0 0 2 2 】

20

また、図 1 に示す例においては、サーバ 1 0 - 1 がマスタノードとして機能し、サーバ 1 0 - 2 ~ 1 0 - 4 がスレーブノードとして機能する。以下、サーバ 1 0 - 1 をマスタノード MN という場合があり、サーバ 1 0 - 2 ~ 1 0 - 4 をそれぞれスレーブノード SN という場合がある。

マスタノード MN は、本分散処理システム 1 における処理を管理する装置であり、複数のスレーブノード SN にタスクをそれぞれ割り当てる。スレーブノード SN は、マスタノード MN から割り当てられたマップタスク（Map タスク；以下、単にタスクという）を実行する。タスクを分散して割り当てられた複数のスレーブノード SN が、それぞれ割り当てられたタスクを並行して実行することによりジョブの処理時間を短縮する。

【 0 0 2 3 】

30

なお、図 1 に示す例においては、マスタノード MN もタスクトラッカ 1 3（後述）としての機能を備えており、このマスタノード MN においても割り当てられたタスクの実行を行なう機能を備えている。従って、図 1 に例示する分散処理システム 1 においては、サーバ 1 0 - 1 もスレーブノード SN として機能する。

また、本実施形態においては、各スレーブノード SN が、それぞれマップタスクを実行するものとする。

【 0 0 2 4 】

サーバ 1 0 は、例えばサーバ機能をそなえたコンピュータ（情報処理装置、データ分割装置、処理部）である。サーバ 1 0 は、図 2 に示すように、CPU（Central Processing Unit）2 0 1 , メモリ 2 0 2 , 表示装置 2 0 5 , キーボード 2 0 6 , マウス 2 0 7 及び記憶装置 2 0 8 を備える。

40

メモリ 2 0 2 は ROM（Read Only Memory）及び RAM（Random Access Memory）を含む記憶装置である。メモリ 2 0 2 の ROM には、分散処理制御に係るソフトウェアプログラムやこのプログラム用のデータ類が書き込まれている。メモリ 2 0 2 上のソフトウェアプログラムは、CPU 2 0 1 に適宜読み込まれて実行される。また、このメモリ 2 0 2 の RAM には、後述するノード間共有情報 1 5 が格納される。

【 0 0 2 5 】

また、メモリ 2 0 2 の RAM は、一次記憶メモリあるいはワーキングメモリとして利用される。さらに、メモリ 2 0 2 の RAM は、分割するファイルを格納する図示しないバッファ領域としても機能する。

50

表示装置 205 は、例えば、液晶ディスプレイや C R T (Cathode Ray Tube) ディスプレイであり、種々の情報を表示する。

【 0026 】

キーボード 206 及びマウス 207 は入力装置であり、利用者がこれらの入力装置を用いて、種々の入力操作を行なう。例えば、マスタノード MN において、利用者は、これらのキーボード 206 やマウス 207 を用いて、例えば、処理対象のファイルの指定や処理内容の指定 (入力) を行なう。

記憶装置 208 は種々のデータやプログラムを格納する記憶装置であって、例えば、HDD (Hard Disk Drive) や SSD (Solid State Drive) である。又、記憶装置 208 は、例えば、複数の HDD を組み合わせて、冗長化された 1 つのストレージとして管理する RAID (Redundant Arrays of Inexpensive Disks) 装置であってもよい。

10

【 0027 】

CPU 201 は、種々の制御や演算を行なう処理装置であり、メモリ 202 に格納されたプログラムを実行することにより、種々の機能を実現する。

マスタノード MN において、CPU 201 は、図 1 に示す、ユーザアプリケーション機能部 11 , ファイル管理部 12 , ジョブトラッカ 14 及びタスクトラッカ 13 として機能する。

【 0028 】

なお、これらのユーザアプリケーション機能部 11 , ファイル管理部 12 , ジョブトラッカ 14 及びタスクトラッカ 13 としての機能を実現するためのプログラムは、例えばフレキシブルディスク, CD (CD - ROM , CD - R , CD - RW 等) , DVD (DVD - ROM , DVD - RAM , DVD - R , DVD + R , DVD - RW , DVD + RW , HDD DVD 等) , ブルーレイディスク, 磁気ディスク, 光ディスク, 光磁気ディスク等の、コンピュータ読取可能な記録媒体に記録された形態で提供される。そして、コンピュータはその記録媒体からプログラムを読み取って内部記憶装置または外部記憶装置に転送し格納して用いる。又、そのプログラムを、例えば磁気ディスク, 光ディスク, 光磁気ディスク等の記憶装置 (記録媒体) に記録しておき、その記憶装置から通信経路を介してコンピュータに提供してもよい。

20

【 0029 】

ユーザアプリケーション機能部 11 , ファイル管理部 12 , ジョブトラッカ 14 及びタスクトラッカ 13 としての機能を実現する際には、内部記憶装置 (本実施形態ではメモリ 202) に格納されたプログラムがコンピュータのマイクロプロセッサ (本実施形態では CPU 201) によって実行される。このとき、記録媒体に記録されたプログラムをコンピュータが読み取って実行してもよい。

30

【 0030 】

同様に、スレーブノード SN においても、CPU 201 が、プログラムを実行することにより、タスクトラッカ 13 及びファイル管理部 12 として機能する。

ユーザアプリケーション機能部 11 は、ユーザからのジョブ依頼を受け付け、マップリデュースジョブ (Map-Reduce ジョブ ; 以下、単にジョブという) を生成し、ジョブトラッカ 14 にジョブを投入 (ジョブ登録) する。

40

【 0031 】

ユーザアプリケーション機能部 11 は、ユーザがキーボード 206 やマウス 207 を用いて処理対象ファイルの指定や処理内容 (指示内容) を入力すると、この入力された情報に基づいてジョブを生成する。

ジョブトラッカ 14 は、ユーザアプリケーション機能部 11 によって行なわれたジョブ登録に基づいて、クラスタ中の利用可能なタスクトラッカ 13 にタスクを割り当てる、割り当て制御部としての機能を備える。

【 0032 】

タスクトラッカ 13 は、ジョブトラッカ 14 から割り当てられたタスクを処理する。

なお、これらのユーザアプリケーション機能部 11 , ジョブトラッカ 14 及びタスク

50

ラッカ 13 としての機能は既知の種々の手法で実現することができ、その詳細な説明は省略する。

ファイル管理部 12 は、処理対象のファイルを複数のサーバ 10 の記憶装置 208 に分散して格納させる。以下、サーバ 10 の記憶装置 208 にデータを格納することを、単にサーバ 10 にデータを格納すると表現する。図 1 に示す例においては、サーバ 10 - 1 にファイル 1 が、サーバ 10 - 2 にファイル 4 が、サーバ 10 - 3 にファイル 2, 5 が、サーバ 10 - 4 にファイル 3 が、それぞれ格納されている。

【0033】

また、ファイル管理部 12 は、処理対象のファイル（データ）を所定サイズ（例えば 64 MB）の断片（データブロック、ブロック）に分割して、各ノードの記憶装置 208 に格納させる。すなわち、ファイル管理部 12 はバイナリデータであるファイルを複数のブロック（部分バイナリデータ）に分割する分割処理部 20 としての機能を備える。また、この分割処理部 20 としての機能を備えるサーバ 10 は、ファイルを複数に分割した分割ファイルを処理するデータ分割処理装置として機能する。以下、分割されたファイルの個々の部分を分割ファイルという場合がある。

【0034】

さらに、分割処理部 20 は、割り当てられた分割ファイルに含まれる全ての可変長レコードが、その途中で分断されることがないように、分割ファイルの区切り位置を調整する機能を備える。

分割処理部 20 は、割り当てられた分割ファイルが可変長レコードの途中で分断されている場合、すなわち、分割ファイルの先頭や終端の区切り位置が可変長レコードの途中に位置する場合には、その区切り位置をずらす。これにより、分割ファイル中に含まれる全ての可変長レコードが、それぞれ 2 つのレコード長とデータ（レコードデータ）とを含むようにする。つまり、分割ファイル中の可変長レコードが連続する 2 つの分割ファイルに分かれて格納される（いわゆる泣き別れ状態）となることがないように、区切り位置を調整する。

【0035】

分割処理部 20 は、図 1 に示すように、ファイル読込部 21, ファイル先頭記録部 22, 特定部 23, ファイル終端記録部 24 及び整合性判定部 25 を備える。

ファイル読込部 21 は、メモリ 202 の図示しないバッファ領域に、ユーザアプリケーション機能部 11 によって指定されたファイルのデータを格納する。また、ファイル読込部 21 は、ファイルのデータをバッファ領域に読み込むに際して、バッファ領域の初期化も行なう。

【0036】

ファイル先頭記録部 22 は、ファイルの先頭位置を示す情報をノード間共有情報 15 に記録する。

ノード間共有情報 15 は、サーバ 10（ノード）間で共有される情報であって、各サーバ 10 が、自らに割り当てられた分割ファイル（タスク）について区切り位置の調整後の先頭位置と終端位置とをそれぞれ記録する。

【0037】

図 3 は実施形態の一例としての分散処理システム 1 におけるノード間共有情報 15 を例示する図である。

この図 3 に示す例においては、ノード間共有情報 15 は、タスク番号に対して開始位置及び終端位置を関連付けて構成されている。

タスク番号は、サーバ 10 に割り当てられるタスクを特定する識別情報であり、タスクとは分割ファイルを示す。図 3 に示す例においてはタスク番号として 1 ~ 3 の整数が用いられている。ただし、タスク番号は整数に限定されるものではなく、任意の文字列等を用いてもよく、種々変形して実施することができる。なお、以下、タスク番号のことをタスク ID という場合がある。

【0038】

開始位置は分割ファイルの開始位置を示す情報であり、終端位置は分割ファイルの終端位置を示す情報である。これらの開始位置及び終了位置としては、例えば、分割前のファイルの先頭位置を基準（例えば0）とするアドレス情報が用いられる。

このように、ノード間共有情報15を格納するメモリ202は、検出した区切り位置を示す情報（確定レコード先頭位置、確定レコード終端位置）を記憶する記憶手段として機能する。また、このノード間共有情報15に開始位置（確定レコード先頭位置）や終端位置（確定レコード終端位置）を記憶するファイル先頭記録部22やファイル終端記録部24は、検出した区切り位置を示す情報をメモリ202に登録する登録部として機能する。

【0039】

なお、ノード間共有情報15は、例えば、マスタノードMNのメモリ202に格納され、各サーバ10はこのマスタノードMNに格納されたノード間共有情報15に対して更新や参照を行なう。すなわち、各サーバ10がノード間共有情報15を共用する。

ただし、ノード間共有情報15の格納場所はマスタノードMNに限定されるものではなく、種々変形して実施することができる。例えば、いずれかのスレーブノードSNにノード間共有情報15を格納してもよく、又、図示しない他の情報処理装置がノード間共有情報15を格納してもよい。また、各サーバ10が、それぞれノード間共有情報15を備え、サーバ間通信を行なうことで各サーバ10が有するノード間共有情報15を同期させてもよい。

【0040】

ファイル先頭記録部22は、後述する特定部23により特定される分割ファイル（タスク）の先頭位置を、ノード間共有情報15にその分割ファイルのタスク番号に対応付けて開始位置として記録する。

ファイル終端記録部24は、特定部23が特定した分割ファイルの終端位置（確定レコード終端位置）を示す情報を、ノード間共有情報15に終端位置として記録する。

【0041】

特定部23は分割ファイルのデータ構成上の特性（規則性）に基づき、分割ファイルに含まれる可変長レコードの位置（レコード位置、データ区切り位置）を特定するものであり、分割ファイルを構成する可変長レコード（所定のデータ）の特性に基づき、データ区切り位置を検出する。

特定部23は、可変長レコードのヘッダに含まれるデータ長情報に基づき、当該可変長レコードと、後続する可変長レコードとの境界位置を特定する。すなわち、特定部23は、可変長レコードのヘッダに含まれるデータ長情報に基づき、当該可変長レコードの位置を特定する。つまり、特定部23は、可変長レコードにおいては、データの前後に当該データのデータ長を示すレコード長情報が配置されているというデータ構造上の特性に基づき、可変長レコードの位置を特定する。

【0042】

特定部23がバッファ領域に格納された分割ファイルをその先頭から走査し、この分割ファイル内において、先頭の可変長レコードの位置を特定すると、当該可変長レコードに後続する各可変長レコードについては、そのデータ長を順次参照することでそれぞれの位置を特定することができる。従って、分割ファイルに含まれる複数の可変長レコードのうち、その先頭の可変長レコードの前側レコード長情報の開始位置を特定することにより、当該分割ファイルに含まれる全ての可変長レコードの位置を特定することができる。これにより、分割ファイルの最後尾に位置する可変長レコードの終端位置も特定することができる。すなわち、特定部23は、バッファ領域に格納された分割ファイルについて、当該分割ファイルに含まれる各可変長レコードを順次認識することで、その終端位置を特定する。

【0043】

なお、バッファ領域における分割ファイルの走査位置は、例えば、図示しないポインタにより特定することができる。又、バッファ領域において、分割ファイルの先頭から終端に向かう方向を走査方向という場合がある。

10

20

30

40

50

特定部 2 3 は、例えば、以下の (A 1) ~ (A 6) の処理を行なうことで、分割ファイルの区切り位置を特定する。

【 0 0 4 4 】

図 4 は実施形態の一例としての分散処理システム 1 における特定部 2 3 による区切り位置の特定方法を説明するための図である。なお、図 4 中においては、便宜上、マップタスク 2 について示し、マップタスク 1 , 3 についての図示を省略する。

(A 1) 特定部 2 3 は、分割ファイルが格納されたバッファ領域の先頭から 4 バイトを読み出す。処理対象のファイル中における、この読み出した 4 バイトのデータ位置を仮先頭位置とする。

【 0 0 4 5 】

(A 2) 特定部 2 3 は、4 バイトの値 (X) を解析して、このデータがレコード長であるか否かを確認する。

具体的には、特定部 2 3 は、まず、取得した 4 バイトの値 (X) を符号なし整数値として、 $0 < X < 32768$ の範囲内かどうか確認する。ここで、X の上限値として 32767 と比較するのは、C O B O L のレコード長の最大値が 32767 だからである。

【 0 0 4 6 】

4 バイトの値が $0 < X < 32768$ の範囲外の場合は、この 4 バイトの値 (X) はレコード長を表すものではないと判断できる。この場合、特定部 2 3 は、バッファ領域において読み出しの先頭位置を走査方向に 1 バイトずらして、新たな 4 バイトのデータを読み出し、再度 (A 2) の処理を実施する。4 バイトの値が $0 < X < 32768$ の範囲内の場合は、(A 3) の処理に移行する。

【 0 0 4 7 】

(A 3) 取得した 4 バイトの値 (図 4 に示す例では “ 2 0 0 0 ”) がレコード長情報であると仮定して、バッファ領域において、走査方向に、このレコード長情報が示すレコード長分進んだ位置 (バッファ位置) に、同一の値のレコード長情報 (後側レコード長情報) が存在するか確認する。

すなわち、バッファ領域において、走査方向に、このレコード長分進んだ位置から 4 バイトのデータを読み出し、先に取得した 4 バイトの値と同一であるか否かを確認する。

【 0 0 4 8 】

レコード長 (2 0 0 0 バイト) だけ進んだバッファ位置に同じレコード長 (“ 2 0 0 0 ”) が存在しない場合は、取得した 4 バイトの値はレコード長ではなかったと判断され、(A 2) の処理に戻る。一方、レコード長 (2 0 0 0 バイト) だけ進んだバッファ位置に同じレコード長 (“ 2 0 0 0 ”) が存在する場合は、レコード位置が検出できたものとして (A 4) の処理に移行する。

【 0 0 4 9 】

(A 4) 取得した 4 バイトの値がレコード長であると想定し、そのレコード先頭位置 (確定レコード先頭位置) をファイル先頭記録部 2 2 に通知する。

(A 5) ファイル先頭記録部 2 2 は、通知された確定レコード先頭位置を、当該分割ファイルに相当するタスク番号に関連づけてノード間共有情報 1 5 に記録する。

また、正しいレコードの開始位置が判明したものとしてレコードの読み込み処理を行なう。すなわち、分割ファイルを構成する各可変長レコードのデータを、先頭から順次読み出す。この際、各可変長レコードについて、ヘッダに含まれるレコード長とデータのレコード長との整合性を確認しながら読み込みを行なう。

【 0 0 5 0 】

不整合が検出された場合には、上記 (A 3) ~ (A 5) における処理結果を破棄し、バッファ領域において読み出しの先頭位置を 1 バイトずらして、新たな 4 バイトのデータを読み出し、(A 2) の処理からやり直す。

(A 6) 解析対象の分割ファイルの終端位置まで到達したら、特定部 2 3 は、その終端位置を示す情報を確定レコード終端位置としてファイル終端記録部 2 4 に通知する。

【 0 0 5 1 】

10

20

30

40

50

上述した確定レコード先頭位置及び確定レコード終端位置が、当該分割ファイルの区切り位置として用いられる。

従って、特定部 2 3 は、処理対象の分割ファイル（部分バイナリデータ）のデータ区切り位置を、バイナリデータを構成する可変長レコードの構造上の特性に基づき検出する検出部として機能する。

【 0 0 5 2 】

その後、後述する整合性判定部 2 5 が、ノード間共有情報 1 5 を参照し、他のマップタスクで検出し記録した分割ファイルの位置との整合性チェックを行なう。

整合性判定部 2 5 は、特定部 2 3 によって特定された分割ファイルの位置と、他のマップタスクにおいて特定された分割ファイルの位置とを比較することにより、特定した分割ファイルの区切り位置の整合性を判定する。

10

【 0 0 5 3 】

整合性判定部 2 5 は、ノード間共有情報 1 5 を参照することで、特定部 2 3 によって特定された確定レコード先頭位置と、他のマップタスクで検出し記録した分割ファイルの終端位置との整合性を確認する。

具体的には、整合性判定部 2 5 は、特定部 2 3 によって特定された確定レコード先頭位置を、処理対象のファイルにおいて、当該分割ファイルよりも前側（先行して）に隣接する他の分割ファイル进行处理するマップタスクについてノード間共有情報 1 5 に記録されている分割ファイルの終端位置（確認レコード終端位置）と比較する。この比較の結果、先行する分割ファイルの確認レコード終端位置が、当該マップタスクにおいて特定される確定レコード先頭位置と連続している場合に、整合性判定部 2 5 は整合性があると判断する。

20

【 0 0 5 4 】

従って、整合性判定部 2 5 は、メモリ 2 0 2 に登録された他の分割ファイル（部分バイナリデータ）に関する区切り位置を示す情報に基づき、検出した区切り位置が正当であることを判定する判定部として機能する。

整合性判定部 2 5 により整合性があると判定されると、分割処理部（確定部）2 0 は、確定レコード先頭位置と確定レコード終端位置とを分割ファイルのデータ区切り位置として確定する。すなわち、分割ファイルのデータ範囲は、確定レコード先頭位置が先頭であり、且つ、確定レコード終端位置が終端となるよう修正される。

30

【 0 0 5 5 】

従って、分割処理部 2 0 は、判定されたデータ区切り位置に基づき処理対象のファイル（バイナリデータ）の区切り位置を確定する確定部として機能する。

このようにファイル管理部 1 2（分割処理部 2 0）によりデータの区切り位置が修正された分割ファイルは、タスクトラッカ 1 3 に受け渡され、処理される。

上述の如く構成された実施形態の一例としての分散処理システム 1 における分割ファイルの区切り位置の調整方法を、図 5 に示すフローチャート（ステップ B 1 ～ B 1 2）に従って説明する。

【 0 0 5 6 】

ステップ B 1 において、ファイル読込部 2 1 は、メモリ 2 0 2 の図示しないバッファ領域に、ユーザアプリケーション機能部 1 1 によって指定されたファイルのデータを格納する。

40

このステップ B 1 における、ファイル読込部 2 1 によるファイルの読み込み処理を、図 6 に示すフローチャート（ステップ B 1 0 1 ～ B 1 0 2）に従って説明する。

【 0 0 5 7 】

ファイル先頭記録部 2 2 は、ステップ B 1 0 1 においてメモリ 2 0 2 のバッファ領域を初期化した後、ステップ B 1 0 2 において、ユーザアプリケーション機能部 1 1 によって指定されたファイルからバッファ単位でバッファ領域にデータを読み込む。

次に、図 5 のステップ B 2 において、ファイル先頭記録部 2 2 が、ファイルの先頭位置を示す情報をノード間共有情報 1 5 に記録する。

50

【 0 0 5 8 】

このステップ B 2 における、ファイル先頭記録部 2 2 による処理を、図 7 に示すフローチャート（ステップ B 2 0 1 ~ B 2 0 3 ）に従って説明する。

先ず、ステップ B 2 0 1 において、ファイル先頭記録部 2 2 は、処理中のタスク（分割ファイル）を特定するタスク ID を取得する。このタスク ID は、例えば、ユーザアプリケーション機能部 1 1 から取得してもよく、又、ファイル管理部 1 2 等が任意に設定してもよい。

【 0 0 5 9 】

ステップ B 2 0 2 において、ファイル先頭記録部 2 2 は、ノード間共有情報 1 5 に接続して、ステップ B 2 0 3 において、タスク ID と処理対象の分割ファイルの先頭の位置を示す情報とをノード間共有情報 1 5 に記録する。

10

次に、図 5 のステップ B 3 において、ファイル読込部 2 1 は、処理対象の分割ファイルにおける終端位置であるか否かを確認する。すなわち、バッファ領域に展開された処理対象の分割ファイルにおいて、処理中（走査中）位置を示すポインタが分割ファイルの終端に到達しているか否かを確認する。

【 0 0 6 0 】

確認の結果、分割ファイルの終端に到達していない場合には（ステップ B 3 の N o ルート参照）、ステップ B 4 において、特定部 2 3 は、ポインタ位置から 4 バイトの値を読み出し、この 4 バイトの値に対して終端位置の判定処理を開始する。

ステップ B 5 において、取得した 4 バイトの値（X）を符号なし整数値として、 $0 < X < 32768$ の範囲内かどうか確認する。

20

【 0 0 6 1 】

4 バイトの値が $0 < X < 32768$ の範囲内の場合は（ステップ B 5 の Y e s ルート参照）、ステップ B 6 において、取得した 4 バイトの値がレコード長（前側レコード長情報）であると仮定して、バッファ領域において、このレコード長分進んだ位置（バッファ位置）から 4 バイトの値を読み出す。

ステップ B 7 において、ステップ B 4 で取得した 4 バイトの値と、ステップ B 6 でレコード長だけ進んだバッファ位置から取得した、後側レコード長情報と仮定される 4 バイトの値とが同一であるか否かを確認する。

【 0 0 6 2 】

30

ステップ B 4 で取得した 4 バイトの値と、ステップ B 6 で取得した 4 バイトの値とが同一である場合には（ステップ B 7 の Y e s ルート参照）、ステップ B 8 に移行する。すなわち、ステップ B 4 において、可変長レコードにおけるレコード長を正しく検出できたものとして、その可変長レコードからデータ（レコードデータ）を読み出し、ステップ B 3 に戻る。

【 0 0 6 3 】

一方、ステップ B 5 において 4 バイトの値が $0 < X < 32768$ の範囲外の場合や（ステップ B 5 の N o ルート参照）、ステップ B 4 で取得した 4 バイトの値と、ステップ B 6 で取得した 4 バイトの値とが同一でない場合には（ステップ B 7 の N o ルート参照）、ステップ B 9 において、分割ファイルにおけるポインタ位置（現在位置）を走査方向に 1 バイト進めた後、ステップ B 4 に戻る。

40

【 0 0 6 4 】

また、ステップ B 3 において、分割ファイルの終端に到達した場合には（ステップ B 3 の Y e s ルート参照）、ステップ B 1 0 において、整合性判定部 2 5 による終端位置に対する判定（検証）とファイル終端記録部 2 4 による終端位置の記録が行なわれる。

このステップ B 1 0 における、整合性判定部 2 5 及びファイル終端記録部 2 4 による処理を、図 8 に示すフローチャート（ステップ B 1 1 1 ~ B 1 1 6 ）に従って説明する。

【 0 0 6 5 】

ステップ B 1 1 1 において、整合性判定部 2 5 は、処理中のタスク（分割ファイル）を特定するタスク ID を取得する。

50

ステップ B 1 1 2 において、整合性判定部 2 5 はノード間共有情報 1 5 にアクセスして、ステップ B 1 1 3 において、ファイル終端記録部 2 4 が、ノード間共有情報 1 5 にタスク ID 及び確定レコード終端位置を記録する。

【 0 0 6 6 】

ステップ B 1 1 4 において、整合性判定部 2 5 は、ステップ B 1 1 1 において取得したタスク ID が 1 であるか否かを確認する。この確認の結果、タスク ID が 1 でない場合には (ステップ B 1 1 4 の N o ルート参照) ステップ B 1 1 5 において、整合性判定部 2 5 は、ノード間共有情報 1 5 から、処理対象ファイルにおいて、処理中の分割ファイルよりも前側において連続する (すなわち、一つ前の) 他の分割ファイルについての、確定レコード終端位置を読み出す。

10

【 0 0 6 7 】

そして、ステップ B 1 1 6 において、整合性判定部 2 5 は、この取得した一つ前の分割ファイルについての確定レコード終端位置の値に 1 を加算した値と、当該処理対象ファイルについて特定部 2 3 が特定した確定レコード先頭位置とを比較する。

取得した一つ前の分割ファイルについての確定レコード終端位置の値に 1 を加算した値と、当該処理対象ファイルについて特定部 2 3 が特定した確定レコード先頭位置とが一致した場合には (ステップ B 1 1 6 の Y e s ルート参照)、整合性があると判断して処理を終了する。

【 0 0 6 8 】

一方、取得した一つ前の分割ファイルについての確定レコード終端位置の値に 1 を加算した値と、当該処理対象ファイルについて特定部 2 3 が特定した確定レコード先頭位置とが一致しない場合には (ステップ B 1 1 6 の N o ルート参照)、整合性がないと判断して処理を終了する。

20

なお、整合性の判断結果 (整合性ありもしくは整合性なし) は、その旨を示す情報 (例えばフラグ) をメモリ 2 0 2 等の所定の記憶領域に記録することが望ましい。

【 0 0 6 9 】

また、ステップ B 1 1 4 における確認の結果、タスク ID が 1 である場合には (ステップ B 1 1 4 の Y e s ルート参照)、整合性判定部 2 5 は、整合性があると判断して処理を終了する。ファイルの先頭位置については、先行する分割ファイルは存在しないからである。

30

その後、図 5 におけるステップ B 1 1 において、特定部 2 3 は、整合性判定部 2 5 による判定結果を参照して、整合性があると判断された場合には (ステップ B 1 1 の Y e s ルート参照)、処理を終了する。

【 0 0 7 0 】

また、整合性がないと判断された場合には (ステップ B 1 1 の N o ルート参照)、ステップ B 1 2 において、特定部 2 3 は、分割ファイルにおけるポインタ位置 (現在位置、確定レコード先頭位置) を走査方向に 1 バイトずらし、ステップ B 3 に戻る。

次に、上述の如く構成された実施形態の一例としての分散処理システム 1 における解析対象ファイルの分割処理を、図 9 ~ 図 1 2 を用いて例示する。図 9 は解析対象ファイル F を例示する図である。解析対象ファイル F は、N e t C O B O L の可変長レコード順ファイルであり、以下においては、この図 9 に示すように、ファイル管理部 1 2 が、この解析対象ファイル (ファイル) F を、3 つの分割ファイル D F 1 ~ D F 3 に分割する例を示す。各分割ファイル D F 1 ~ D F 3 はそれぞれ 6 4 M B のデータサイズを有する。

40

【 0 0 7 1 】

図 1 0 , 図 1 1 及び図 1 2 は、それぞれ分割ファイル D F 1 , D F 2 及び D F 3 に対する処理を示す。

分割ファイル D F 1 はマップタスク 1 に (図 1 0 参照)、分割ファイル D F 2 はマップタスク 2 に (図 1 1 参照)、また、分割ファイル D F 3 はマップタスク 3 に (図 1 2 参照)、それぞれ割り当てられ、各マップタスク 1 ~ 3 によりそれぞれ処理される。

【 0 0 7 2 】

50

また、各マップタスク 1 ~ 3 において、割り当てられた分割ファイル D F 1 ~ D F 3 は、それぞれに含まれる全ての可変長レコードが、それぞれ途中で分断されることがないように、上述した手法により区切り位置が調整される。

また、その際、各マップタスクにおいては、ファイル先頭記録部 2 2 が、区切り位置の調整後の分割ファイルの確定レコード先頭位置をノード間共有情報 1 5 の開始位置に格納し、又、確定レコード終端位置を、ノード間共有情報 1 5 の終端位置に格納する。

【 0 0 7 3 】

図 1 0 に例示するマップタスク 1 において処理される分割ファイル D F 1 は、解析対象ファイル F の先頭部分であるので、その確定レコード先頭位置は 0 であり、ノード間共有情報 1 5 のタスク番号 1 の開始位置には 0 が登録される。

10

また、区切り位置の調整後の分割ファイル D F 1 の終端位置が $64\text{MB} + 904\text{B}$ (バイト) であり、ノード間共有情報 1 5 のタスク番号 1 の終端位置には $64\text{MB} + 904\text{B}$ が登録される。

【 0 0 7 4 】

図 1 1 に例示するマップタスク 2 において処理される分割ファイル D F 2 についても同様に、区切り位置の調整後の分割ファイル D F 2 の確定レコード先頭位置は $64\text{MB} + 905\text{B}$ であり、ノード間共有情報 1 5 のタスク番号 2 の開始位置には $64\text{MB} + 905\text{B}$ が登録される。

また、区切り位置の調整後の分割ファイル D F 2 の終端位置は $128\text{MB} + 381\text{B}$ であり、ノード間共有情報 1 5 のタスク番号 2 の終端位置には $128\text{MB} + 381\text{B}$ が登録される。

20

【 0 0 7 5 】

図 1 2 に例示するマップタスク 3 において処理される分割ファイル D F 3 についても同様に、区切り位置の調整後の分割ファイル D F 3 の確定レコード先頭位置は $128\text{MB} + 382\text{B}$ であり、ノード間共有情報 1 5 のタスク番号 3 の開始位置には $128\text{MB} + 382\text{B}$ が登録される。

また、区切り位置の調整後の分割ファイル D F 3 の終端位置は $128\text{MB} + 2079\text{B}$ であり、ノード間共有情報 1 5 のタスク番号 3 の終端位置には $128\text{MB} + 2079\text{B}$ が登録される。

【 0 0 7 6 】

30

このように、実施形態の一例としての分散処理システム 1 によれば、各マップタスクにおいて、特定部 2 3 が、バイナリデータである分割ファイルを、そのデータ構成の規則性を用いることで、その区切り位置を特定する。

例えば、分割ファイルを構成する可変長レコードが、データの前後にデータ長を示す情報を有している場合に、バッファ領域に格納された分割ファイルの先頭から走査して、レコード長を示すデータ (前側データ長情報) と想定される部分を検出する。そして、このレコード長と想定されるデータが示す値 (データ長) だけ進んだバッファ位置にも、同一のレコード長と想定されるデータ (後側データ長情報) が検出された場合に、可変長レコードのレコード位置が検出できたとみなす。

【 0 0 7 7 】

40

この可変長レコードのレコード位置の検出を、分割ファイルの終端まで繰り返し行ない、また、途中で不整合が検出された場合には、開始位置を 1 バイト進めた位置を新たな確定レコード先頭位置としてレコード位置の検出を再度行なう。これにより、データ区切りを検証することができる。

本分散処理システム 1 によれば、処理対象ファイル全体をその先頭から可変長レコードのレコード長を順次取り出して加算することなく、分割ファイルの区切り位置 (分断位置) を容易に特定することができる。従って、マップタスク処理に要する時間を短縮することができ、処理を高速化することができる。

【 0 0 7 8 】

また、分割ファイルのデータ範囲を、確定レコード先頭位置を先頭とし、確定レコード

50

終端位置を終端とするように修正することで、可変長レコードの途中にデータ区切りが位置することがなく、ブロック分割による泣き別れを防止することができる。

このように、処理対象のデータがバイナリデータであっても、データの規則性を利用し、且つ、規則性によるデータ区切りを検証することで、ブロック分割による泣き別れを防止することができる。

【 0 0 7 9 】

また、各マップタスクにおいて、検証されたデータ区切りに基づき、その確定レコード先頭位置と確定レコード終端位置とをノード間共有情報 1 5 に記録する。そして、整合性判定部 2 5 が、このノード間共有情報 1 5 を参照して、隣り合う分割ファイルの終端位置が、確定レコード先頭位置と連続することを確認する。

10

分割ファイルがバイナリデータであるので、可変長レコードのデータ部分に、レコード長データに似たデータが存在するおそれがある。特定部 2 3 による区切り位置の検証時に、データ中のレコード長に似た値をレコード長と誤認し、更に、そのレコード長に似た値が示すデータ長だけ進んだ位置に、そのレコード長に似た値と同じ値が存在する場合に、区切り位置が誤検知されるおそれがある。

【 0 0 8 0 】

図 1 3 は実施形態の一例としての分散処理システム 1 の特定部 2 3 が犯しうる区切り位置の誤検知を例示する図である。

分割ファイルが格納されたバッファ領域から読み出した 4 バイトの値が 32768 未満の整数値であると、特定部 2 3 は、この値をレコード長と誤検知することになる。なお、図 1 3 中においては、便宜上、マップタスク 2 についてのみ示し、他のマップタスク 1 , 3 について図示を省略する。

20

【 0 0 8 1 】

図 1 3 に示す例では、バッファ領域から読み出した 4 バイトの値が “ 1 0 0 ” であったとする（符号 C 1 参照）。なお、この図 1 3 に示す例においては、本来のデータ長は、図中で “ 1 0 0 0 ” が記載されている部分である。

特定部 2 3 は、取得した 4 バイトの値（ “ 1 0 0 ” ）がレコード長であると仮定して、バッファ領域において、このレコード長分進んだ位置（バッファ位置）に、同一のレコード長データが存在するか確認する。

【 0 0 8 2 】

30

そして、この確認の結果、1 0 0 バイト進んだ先の 4 バイトの領域に、偶然、同じく “ 1 0 0 ” が格納されている場合に（符号 C 2 参照）、特定部 2 3 は、レコード位置が検出できたものと判断してしまう。また、これにより、可変長レコードからデータ部分とみなされるデータの読み込みが行なわれる（符号 C 3 参照）。このようにして読み込まれたデータ中は、本来レコード長を示すデータ “ 1 0 0 0 ” が含まれる等、正しいデータではない。

【 0 0 8 3 】

また、当該分割ファイルにおけるそれ以降の部分について、各可変長レコードについて、ヘッダに含まれるレコード長とデータのレコード長との整合性を確認しながら読み込みを行なった結果、その終端まで不整合が検出されなかった場合には（符号 C 4 参照）、読み込まれた誤ったデータが使用されることになる。

40

このような場合に、特定部 2 3 が確定レコード先頭位置及び確定レコード終端位置をノード間共有情報 1 5 に記録し、整合性判定部 2 5 が、他のノードによってノード間共有情報 1 5 に記録された、当該分割ファイルと連続する他の分割ファイルの確定レコード先頭位置や確定レコード終端位置との連続性を調査することで、特定部 2 3 による誤検知を検知することができるのである（符号 C 5 参照）。

【 0 0 8 4 】

このように、整合性判定部 2 5 がノード間による整合性の確認を行なうことにより、検出したデータ区切りがファイル全体で整合性を有することを確認することができ、信頼性を向上させることができる。

50

そして、開示の技術は上述した実施形態に限定されるものではなく、本実施形態の趣旨を逸脱しない範囲で種々変形して実施することができる。本実施形態の各構成及び各処理は、必要に応じて取捨選択することができ、あるいは適宜組み合わせてもよい。

【0085】

例えば、上述した実施形態においては、整合性判定部25は、分割ファイルの先頭位置が、他のマップタスクで検出し記録した分割ファイルの終端位置と連続するか否かを確認することで整合性を確認しているが、これに限定されるものではない。

すなわち、整合性判定部25は、特定部23によって特定された分割ファイルの終端位置が正しいものであるか否かを検証してもよい。

【0086】

整合性判定部25は、ノード間共有情報15を参照することで、特定部23によって特定された終端位置と、他のマップタスクで検出し記録した分割ファイルの先頭位置との整合性を確認することができる。

具体的には、整合性判定部25は、特定部23が特定した終端位置を、処理対象の分割ファイルに後続する他の分割ファイル进行处理するマップタスクについてノード間共有情報15に記録されている分割ファイルの先頭位置（確認レコード先頭位置）と比較する。この比較の結果、後続する分割ファイルの確認レコード先頭位置が、当該マップタスクにおいて到達した終端位置と連続している場合に、整合性判定部25は整合性があると判断してもよい。

【0087】

また、上述した実施形態においては、処理対象のファイルがNetCOBOLの規格に沿った可変長レコードファイルである例を示しているが、これに限定されるものではなく、種々変形して実施することができる。

図14はMicroFocusのレコード順ファイルの構成を例示する図である。

この図14に例示するレコード順ファイルは、可変長レコードで構成され、各可変長レコードが、書き込まれたデータ（レコード）の前に、レコード長を含むレコードヘッダーを備えるとともに、データの後に最大3文字の埋め込み文字を備える。埋め込み文字は、次のレコードが4バイトの境界で開始されるようにするためのものである。

【0088】

また、このレコード順ファイルには、128バイトのファイルヘッダーレコードが含まれる。

本分散処理システム1においては、処理対象ファイルがこの図14に例示するようなレコード順ファイルであってもよい。すなわち、特定部23は、そのデータ構造の規則性に基づいて、分割ファイルに含まれる可変長レコードの位置（データ区切り位置）を特定する。

【0089】

具体的には、特定部23は、バッファ領域に格納された分割ファイルにおいて、ヘッダに含まれるレコード長を読み取り、このレコード長分進んだ位置から所定サイズ（最大3文字）のデータを読み出し、このデータが埋め込み文字であるか否かを確認すればよい。

このように、処理対象のデータ構造に応じて、特定部23が、可変長レコードからのデータ読出位置や比較対象を適宜変更して実施することで、種々の処理対象データに適用することができる。

【0090】

また、上述した実施形態においては、複数のサーバ10を備え、これらの各サーバ10においてそれぞれマップタスクが実行されるとともに分割処理部20が分割ファイルの区切り位置の特定（修正）を行なっているが、これに限定されるものではない。例えば、1つのサーバ（情報処理装置）10において、複数のマップタスクを実行してもよく、これらのマップタスク毎に分割処理部20としてのプロセスを実行して分割ファイルの区切り位置の特定（修正）を行なってもよい。

【0091】

以上の実施形態に関し、更に以下の付記を開示する。

(付記 1)

コンピュータに、

処理対象である、バイナリデータが複数に分割された部分バイナリデータのデータ区切り位置を、前記バイナリデータを構成する所定のデータの特性に基づき検出し、

検出したデータ区切り位置を示す情報を記憶手段に登録し、

前記記憶手段に登録された他の部分バイナリデータに関するデータ区切り位置を示す情報に基づき、前記部分バイナリデータにおいて検出したデータ区切り位置が正当であるかを判定する

処理を実行させることを特徴とする、データ分割処理プログラム。

10

【0092】

(付記 2)

前記処理対象の部分バイナリデータについて検出したデータ区切り位置と、前記記憶手段に登録された、前記バイナリデータにおいて前記処理対象の部分バイナリデータと隣接する部分バイナリデータについてのデータ区切り位置とが連続する場合に、検出したデータ区切り位置が正当であると判断する

処理を前記コンピュータに実行させることを特徴とする、付記 1 記載のデータ分割処理プログラム。

【0093】

(付記 3)

前記部分バイナリデータを構成する個々の単位レコードに含まれるデータ長情報に基づき、当該部分バイナリデータの位置を特定する

処理を前記コンピュータに実行させることを特徴とする、付記 1 又は 2 記載のデータ分割処理プログラム。

20

【0094】

(付記 4)

正当であると判断された前記データ区切り位置に基づき、前記バイナリデータのデータ区切り位置を確定する

処理を前記コンピュータに実行させることを特徴とする、付記 1 ~ 3 のいずれか 1 項に記載のデータ分割処理プログラム。

30

【0095】

(付記 5)

バイナリデータを複数に分割した部分バイナリデータを処理するデータ分割処理装置であって、

処理対象の部分バイナリデータのデータ区切り位置を、前記バイナリデータを構成する所定のデータの特性に基づき検出する検出部と、

検出したデータ区切り位置を示す情報を記憶手段に登録する登録部と、

前記記憶手段に登録された他の部分バイナリデータに関するデータ区切り位置を示す情報に基づき、前記部分バイナリデータにおいて検出したデータ区切り位置が正当であるかを判定する判定部と、

を備えることを特徴とする、データ分割処理装置。

40

【0096】

(付記 6)

前記判定部が、

前記処理対象の部分バイナリデータについて検出したデータ区切り位置と、前記記憶手段に登録された、前記バイナリデータにおいて前記処理対象の部分バイナリデータと隣接する部分バイナリデータについてのデータ区切り位置とが連続する場合に、検出したデータ区切り位置が正当であると判断する

ことを特徴とする、付記 5 記載のデータ分割処理装置。

【0097】

50

(付記 7)

前記検出部が、前記部分バイナリデータを構成する個々の単位レコードに含まれるデータ長情報に基づき、当該部分バイナリデータの位置を特定することを特徴とする、付記 5 又は 6 記載のデータ分割処理装置。

(付記 8)

正当であると判断された前記データ区切り位置に基づき、前記バイナリデータのデータ区切り位置を確定する確定部を備えることを特徴とする、付記 5 ～ 7 のいずれか 1 項に記載のデータ分割処理装置。

【0098】

(付記 9)

バイナリデータを複数に分割して複数の処理部により処理する分割処理方法であって、前記複数の処理部のそれぞれに、前記バイナリデータの一部である部分バイナリデータをそれぞれ割り当て、

前記複数の処理部はそれぞれ、割り当てられた部分バイナリデータのデータ区切り位置を、前記バイナリデータを構成する所定のデータの特性に基づき検出するとともに、検出したデータ区切り位置を示す情報を記憶手段に登録し、

前記複数の処理部はそれぞれ、他の処理部により前記記憶手段に登録されたデータ区切り位置を示す情報に基づき、前記部分バイナリデータにおいて検出したデータ区切り位置が正当であるかを判定する

ことを特徴とする、データ分割処理方法。

【0099】

(付記 10)

前記処理対象の部分バイナリデータについて検出したデータ区切り位置と、前記記憶手段に登録された、前記バイナリデータにおいて前記処理対象の部分バイナリデータと隣接する部分バイナリデータについてのデータ区切り位置とが連続する場合に、検出したデータ区切り位置が正当であると判断する

ことを特徴とする、付記 9 記載のデータ分割処理方法。

【0100】

(付記 11)

前記部分バイナリデータを構成する個々の単位レコードに含まれるデータ長情報に基づき、当該部分バイナリデータの位置を特定する

ことを特徴とする、付記 9 又は 10 記載のデータ分割処理方法。

(付記 12)

正当であると判断された前記データ区切り位置に基づき、前記バイナリデータのデータ区切り位置を確定する

ことを特徴とする、付記 9 ～ 11 のいずれか 1 項に記載のデータ分割処理方法。

【符号の説明】

【0101】

1 分散処理システム

10, 10-1～10-4 サーバ(処理部, データ分割処理装置)

11 ユーザアプリケーション機能部

12 ファイル管理部

13 タスクトラッカ

14 ジョブトラッカ

15 ノード間共有情報

20 分散処理部

21 ファイル読込部

22 ファイル先頭記録部(登録部)

23 特定部(検出部, 確定部)

24 ファイル終端記録部(登録部)

10

20

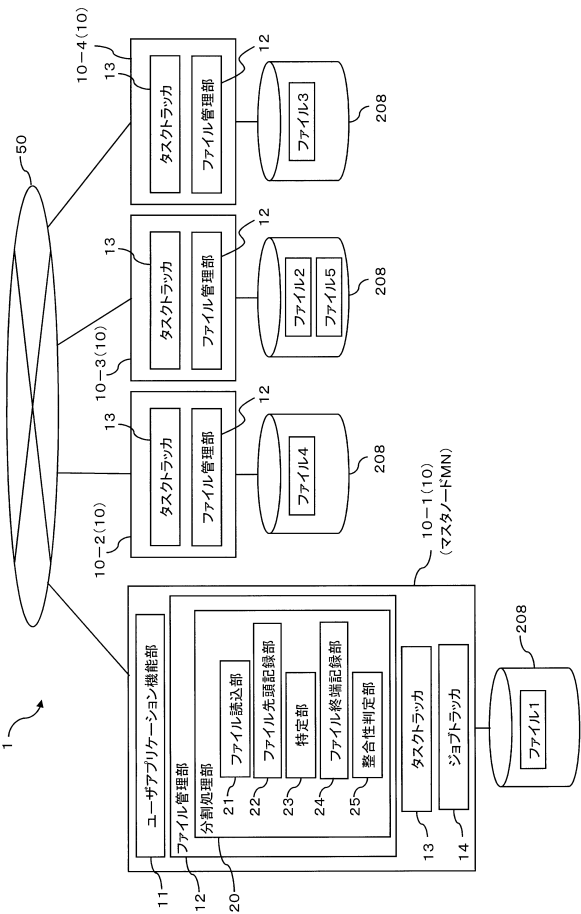
30

40

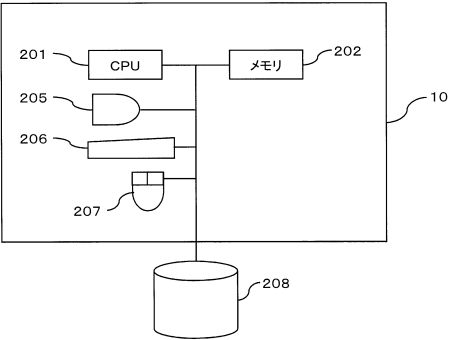
50

- 2 5 整合性判定部（判定部）
- 2 0 1 C P U
- 2 0 2 メモリ（記憶手段）
- 2 0 5 表示装置
- 2 0 6 キーボード
- 2 0 7 マウス
- 2 0 8 記憶装置
- 5 0 ネットワーク

【 図 1 】



【 図 2 】

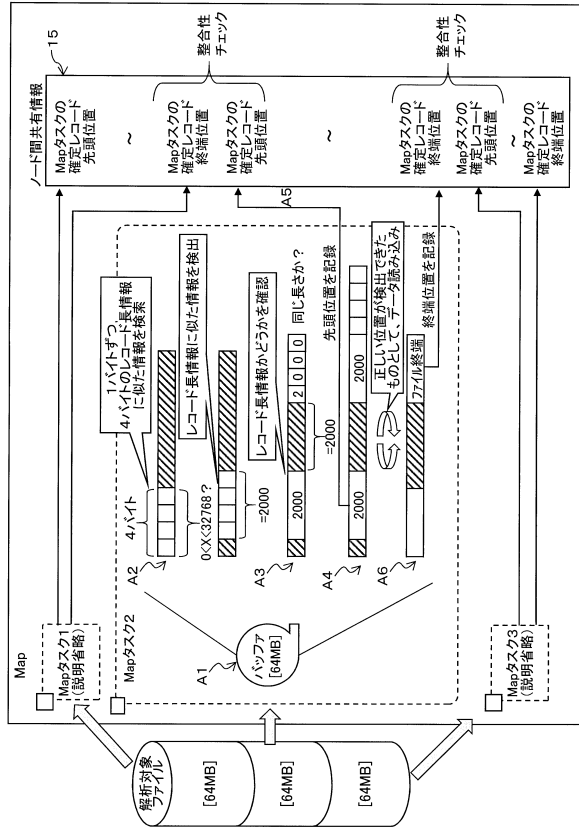


【 図 3 】

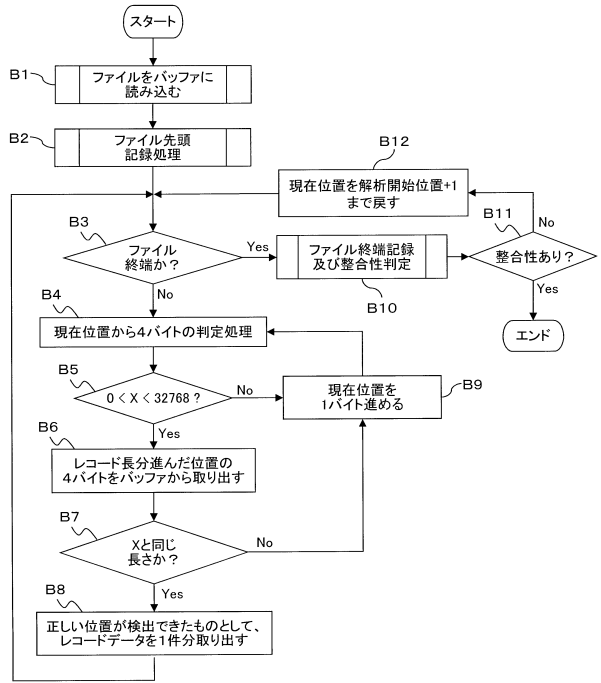
タスク 番号	開始位置	終端位置
1	0	64MB+ 904B
2	64MB+905B	128MB+ 381B
3	128MB+382B	128MB+2079B

15

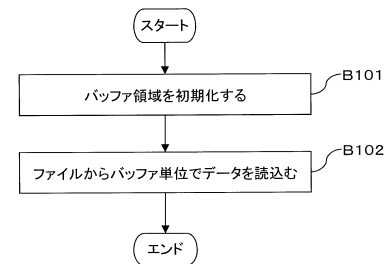
【図4】



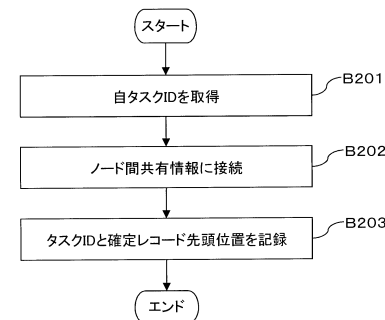
【図5】



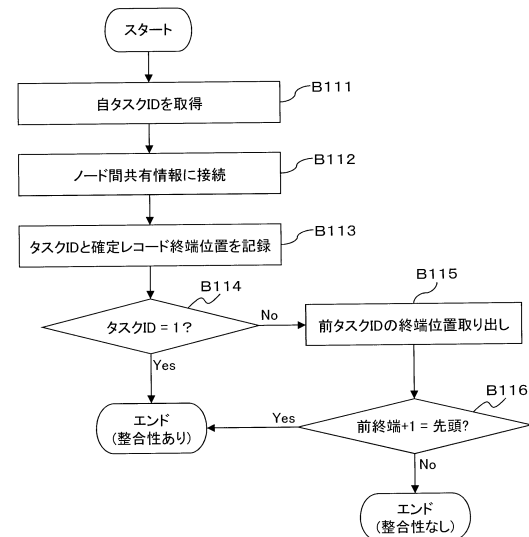
【図6】



【図7】



【図8】

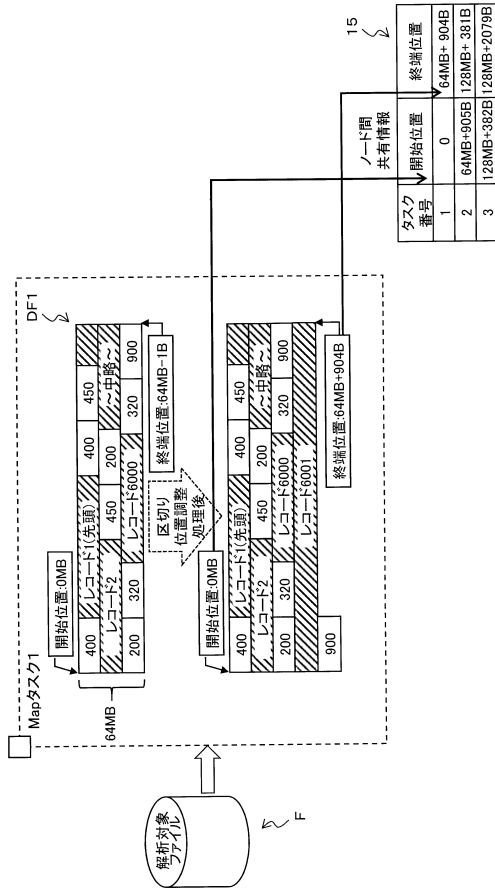


【図9】

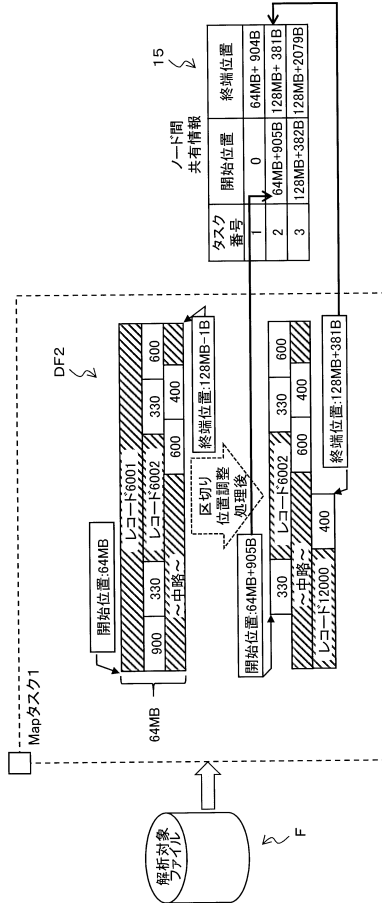


F

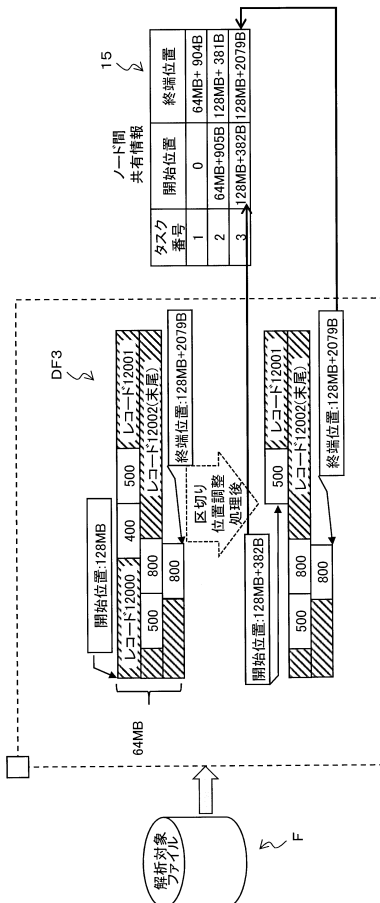
【図 10】



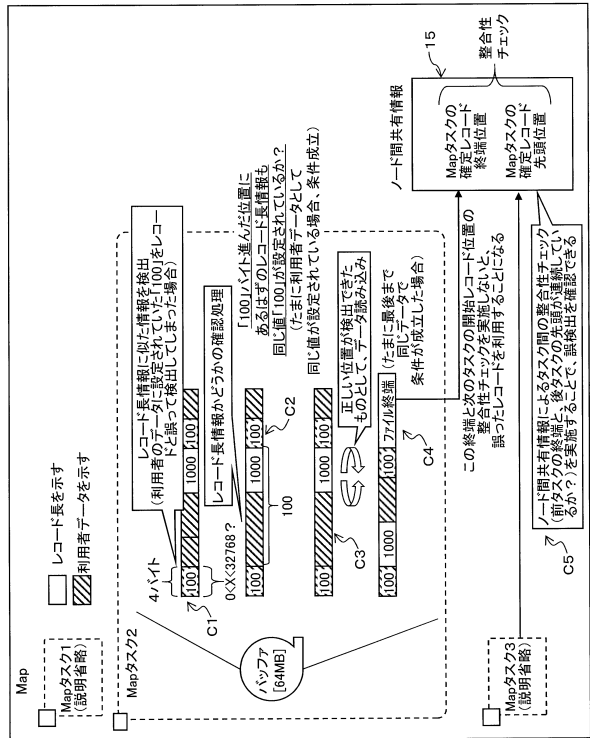
【図 11】



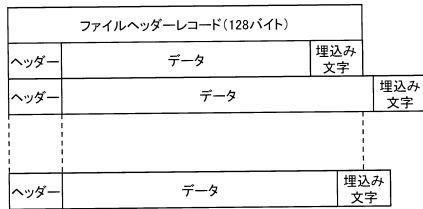
【図 12】



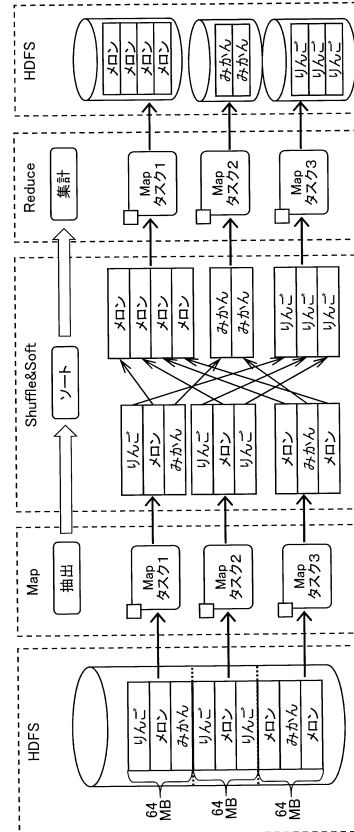
【図 13】



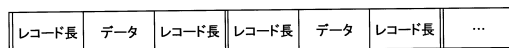
【図 14】



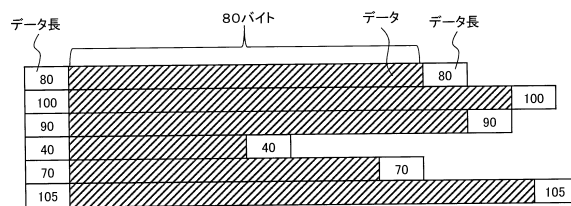
【図 15】



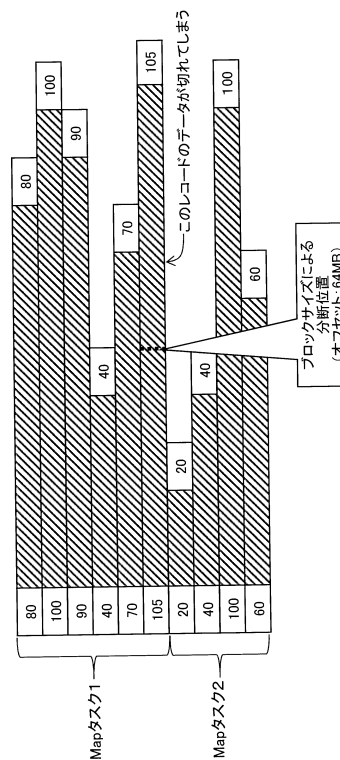
【図 16】



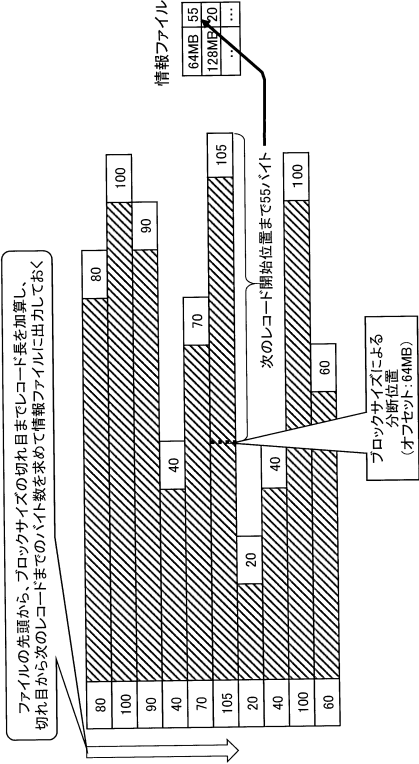
【図 17】



【図 18】



【図 19】



フロントページの続き

- (56)参考文献 特開平 0 3 - 2 1 8 5 6 1 (J P , A)
欧州特許出願公開第 0 4 9 5 9 8 3 (E P , A 1)
特表 2 0 1 0 - 5 1 5 1 1 4 (J P , A)
国際公開第 2 0 0 8 / 0 7 0 4 8 4 (W O , A 2)
米国特許出願公開第 2 0 1 1 / 0 3 0 7 6 5 9 (U S , A 1)
特開 2 0 0 4 - 2 8 7 5 2 5 (J P , A)
国際公開第 2 0 1 4 / 1 8 4 8 5 7 (W O , A 1)
米国特許出願公開第 2 0 1 4 / 0 2 2 9 4 5 2 (U S , A 1)

- (58)調査した分野(Int.Cl. , D B 名)
G 0 6 F 1 2 / 0 0
G 0 6 F 1 7 / 3 0