

(19) 日本国特許庁(JP)

(12) 特許公報(B2)

(11) 特許番号

特許第5230645号
(P5230645)

(45) 発行日 平成25年7月10日(2013.7.10)

(24) 登録日 平成25年3月29日(2013.3.29)

(51) Int.Cl.

F 1

G06F 12/00	(2006.01)	G06F 12/00	510B
G06F 13/00	(2006.01)	G06F 13/00	520A
G06F 17/30	(2006.01)	G06F 17/30	110C
		G06F 17/30	240A

請求項の数 18 (全 18 頁)

(21) 出願番号 特願2009-542017 (P2009-542017)
 (86) (22) 出願日 平成19年12月14日 (2007.12.14)
 (65) 公表番号 特表2010-514035 (P2010-514035A)
 (43) 公表日 平成22年4月30日 (2010.4.30)
 (86) 國際出願番号 PCT/EP2007/064005
 (87) 國際公開番号 WO2008/074751
 (87) 國際公開日 平成20年6月26日 (2008.6.26)
 審査請求日 平成22年11月25日 (2010.11.25)
 (31) 優先権主張番号 11/641,957
 (32) 優先日 平成18年12月20日 (2006.12.20)
 (33) 優先権主張国 米国(US)

(73) 特許権者 508364093
 オーエムエックス テクノロジー エービー
 一
 OMX TECHNOLOGY AB
 スウェーデン国 ストックホルム エスー
 105 78, トゥルヴァクトスヴェー
 ゲン 15
 T u l l v a k t s v a g e n 15,
 S - 1 0 5 7 8 S t o c k h o l m S
 W E D E N
 (74) 代理人 100076428
 弁理士 大塚 康徳
 (74) 代理人 100112508
 弁理士 高柳 司郎

最終頁に続く

(54) 【発明の名称】データセットの変更を最適化するシステム及び方法

(57) 【特許請求の範囲】

【請求項 1】

リモート端末に向けて送信すべき更新データセットを生成するコンピュータシステムであって、前記更新データセットは、ソート済みのデータ要素を含む第1のデータセットとソート済みのデータ要素を含む第2のデータセットとの差分を記述する演算子を含み、前記コンピュータシステムは、

前記第1のデータセット及び前記第2のデータセットを含むメモリと、

前記メモリに接続可能なコンパレータであって、前記第2のデータセット内のデータ要素と前記第1のデータセット内のデータ要素とを順次比較し、先の比較結果が、前記第1のデータセット内のどのデータ要素と前記第2のデータセット内のどのデータ要素が次の比較において比較されるかを制御し、各比較の後に前記メモリ内の変更パラメータを更新し、前記第2のデータセット内のデータ要素が前記第1のデータセット内のデータ要素に一致することを検出するとセレクタを起動するコンパレータと、

前記メモリ内に格納された前記変更パラメータに基づいて演算子を決定し、前記メモリ内に前記決定した演算子を格納する前記メモリ及び前記コンパレータに接続可能なセレクタと

を含み、

前記第2のデータセット及び前記第1のデータセットの差分を記述する演算子を含む前記リモート端末に送信すべき更新データセットを生成することを特徴とするコンピュータシステム。

10

20

【請求項 2】

前記更新データセットを含む更新メッセージを生成及び送信するセレクタに関するコミュニケーション

を更に含むことを特徴とする請求項 1 記載のコンピュータシステム。

【請求項 3】

前記第 1 のデータセット及び前記第 2 のデータセットは、

時間とともに変化する動的データ要素

を含むことを特徴とする請求項 1 記載のコンピュータシステム。

【請求項 4】

前記第 2 のデータセットは、

10

前記第 1 のデータセットよりも新しいバージョンである

ことを特徴とする請求項 1 記載のコンピュータシステム。

【請求項 5】

前記演算子は、

追加演算子、削除演算子、置換演算子を含むグループから決められる
ことを特徴とする請求項 1 記載のコンピュータシステム。

【請求項 6】

前記変更パラメータは、

前記第 1 のデータセットに関する第 1 のカウンタと、前記第 2 のデータセットに関する第 2 のカウンタとを含む

20

ことを特徴とする請求項 1 記載のコンピュータシステム。

【請求項 7】

前記セレクタは、

前記第 1 のデータセットに関する前記第 1 のカウンタと、前記第 2 のデータセットに関する前記第 2 のカウンタとの関係に基づいて演算子を選択する

ことを特徴とする請求項 6 記載のコンピュータシステム。

【請求項 8】

前記変更パラメータは、

前記第 1 のデータセット及び前記第 2 のデータセット内の前記コンパレータの位置の追跡を維持するために、前記第 1 のデータセットに関する第 1 の位置パラメータと、前記第 2 のデータセットに関する第 2 の位置パラメータとを更に含む

30

ことを特徴とする請求項 1 記載のコンピュータシステム。

【請求項 9】

前記演算子は、差分変更を含む

ことを特徴とする請求項 1 記載のコンピュータシステム。

【請求項 10】

各データ要素は、キー及びデータ部分を含む

ことを特徴とする請求項 1 記載のコンピュータシステム。

【請求項 11】

請求項 1 乃至 10 のいずれか 1 項に係わるコンピュータシステムを含む電子商取引システム。

40

【請求項 12】

リモート端末に向けて送信すべき更新データセットを生成する方法であって、前記更新データセットは、ソート済みのデータ要素を含む第 1 のデータセットとソート済みのデータ要素を含む第 2 のデータセットとの差分を記述する演算子を含み、

前記方法は、

前記第 1 のデータセット内のデータ要素と前記第 2 のデータセット内のデータ要素とを順次比較するステップであって、先の比較結果が、前記第 1 のデータセット内のどのデータ要素と前記第 2 のデータセット内のどのデータ要素が次の比較において比較されるかを制御するステップと、

50

各比較の後に、前記第1のデータセット及び前記第2のデータセットを含むメモリ内の変更パラメータを更新するステップと、

前記第2のデータセット内のデータ要素が前記第1のデータセット内のデータ要素に一致することを検出すると、前記メモリに格納された前記変更パラメータに基づいて演算子を決定する選択処理を開始するステップと、

前記決定された演算子を前記メモリに格納するステップと

を含み、

前記第1のデータセット及び前記第2のデータセットの差分を記述する演算子を含む前記リモート端末に送信すべき更新データセットを生成する

ことを特徴とする方法。

10

【請求項13】

前記決定された演算子と比較済みのデータ要素とを関連付けるステップ
を更に含むことを特徴とする請求項12記載の方法。

【請求項14】

追加演算子、削除演算子、置換演算子を含むグループから少なくとも1つの演算子を決定するステップ

を更に含むことを特徴とする請求項12記載の方法。

【請求項15】

前記変更パラメータは、

前記第1のデータセットに関連する第1のカウンタと、前記第2のデータセットに関連する第2のカウンタと

20

を含み、

前記第1のデータセットに関連する前記第1のカウンタと、前記第2のデータセットに関連する前記第2のカウンタとの関係に基づいて演算子を決定するステップ

を更に含むことを特徴とする請求項12記載の方法。

【請求項16】

各データ要素は、キー及びデータ部分を含み、

前記第1のデータセットの要素のキー及びデータ部分の少なくとも1つと、前記第2のデータセットの要素のキー及びデータ部分の少なくとも1つとを比較するステップ

を更に含むことを特徴とする請求項12記載の方法。

30

【請求項17】

前記第1のデータセット及び前記第2のデータセットは、時間とともに変化する動的情報を含む

ことを特徴とする請求項12記載の方法。

【請求項18】

請求項12乃至17のいずれか1項に係わる方法をコンピュータに実行させるコンピュータ・プログラムを格納するコンピュータ可読媒体。

【発明の詳細な説明】

【技術分野】

【0001】

40

本発明は、コンピュータ環境におけるデータ配布に関し、特に、コンピュータネットワーク内の変更の配布用データセットにおける変更の抽出に関する。

【背景技術】

【0002】

今日、ごく一般的に、コンピュータネットワークを介したデータの送信が行なわれている。技術の進歩により送信データ量が急激に増加している。この技術の進歩は、従来に比べてより速い速度でより多くのデータの送信及び処理を可能にしている。また更に、新たなアプリケーションは、より複雑になり、より多くのデータを必要とする。データ配布技術が最も不可欠な部分の一つであるコンピュータシステムの例として、電子商取引システムがある。

50

【0003】

電子商取引における証券、デリバティブ、コモディティやその他の金融商品においては、ユーザに向けて配布する必要のあるデータが大量となっている。これは、取引の決定、統計的計算やその他アセスメントを行なうために、ユーザがデータを必要とするためである。高いパフォーマンスのコンピュータシステムにおいて、この情報全てを抽出及び送信する処理は、プロセッサへの要求が極めて高くなる。CPU時間の量は、回避されうる実行ステップにおいて、無駄にすべきでないわずかなリソースである。

【0004】

更に、中央集権化商取引等に接続するユーザは、一般に、できる限り速く情報の取得を所望する。このようなケースにおいては、例えば、ハードウェアの更新により中央システムのパフォーマンスを上げるだけでは十分ではない場合がある。

10

【0005】

システム内において、ボトルネック又は他のレイテンシの問題を取り除くために、更なる技術が用いられる必要がある。

【0006】

ここで、プロセッサの処理をより効率化するための更なる技術の一つとして、例えば、ユーザ端末側でのデータセットの更新時に種々の手法が存在する。

【0007】

一般的に使用され且つ明らかな解決策としては、古いデータセットを置換する完全な新しいデータセットを常に送信する。この場合、データセットにおけるデータの一部のみが変更される時には非効率である。そのため、より効率的な手法では、変更されるデータセットの部分のみを送信する場合がある。更なる効率化では、差分変更(delta change)を送信する。

20

【0008】

他の周知の技術では、2つのデータセット間の差分を記述する演算子(operator)を送信する。第1のデータセットにおける演算子を適用することにより、当該第1のデータセットを第2のデータセットに変換する。

【0009】

データセットに作用する演算子の良好なセットの選択とともに最適化が行なえる。現在、データセット差分を抽出するのに利用できる方法は質が悪い。

30

【0010】

従って、ステップ数をより少なくする等、効率的な方法で演算子の抽出や選択を行なう技術や、プロセッサにかかる負荷を減らすため、コンピュータシステム内の帯域幅等のデータ配布を減らす技術の構築が必要とされている。

【発明の概要】**【発明が解決しようとする課題】****【0011】**

そこで、本発明は、リモート端末に向けて送信すべき更新データセットを生成するための解決策を提供することを目的とする。

【0012】

本発明は、データセットから効率的にデータを抽出する解決策を提供することを他の目的とする。

40

【0013】

本発明は、データセット間の差分に基づいて、演算子の抽出及び選択の少なくとも1つを効率的に行なう解決策を提供することを他の目的とする。

【0014】

本発明は、リモート端末に向けて送信すべきデータ構成を生成する解決策を提供することを他の目的とする。

【0015】

本発明は、プロセッサ時間をより少なくするのに使用する解決策を提供することを他の

50

目的とする。

【課題を解決するための手段】

【0016】

本発明の第1の側面によれば、上記及び他の目的をコンピュータシステムにより解決する。コンピュータシステムは、リモート端末に向けて送信すべき更新データセットを生成する。更新データセットは、ソート済みのデータ要素を含む第1のデータセットとソート済みのデータ要素を含む第2のデータセットとの差分を記述する演算子を含む。コンピュータシステムは、

第1のデータセット及び第2のデータセットを含むメモリと、

メモリに接続可能なコンパレータであって、第2のデータセット内のデータ要素と第1のデータセット内のデータ要素とを順次比較し、先の比較結果が、第1のデータセット内のどのデータ要素と第2のデータセット内のどのデータ要素が次の比較において比較されるかを制御し、各比較の後にメモリ内の変更パラメータを更新し、第2のデータセット内のデータ要素が第1のデータセット内のデータ要素に一致することを検出するとセレクタを起動するコンパレータと、

メモリ内に格納された変更パラメータに基づいて演算子を決定し、メモリ内に決定した演算子を格納するメモリ及びコンパレータに接続可能なセレクタと

を含み、

第2のデータセット及び第1のデータセットの差分を記述する演算子を含むリモート端末に送信すべき更新データセットを生成する。

【0017】

コンピュータシステムは、商取引システム等のコンピュータシステムが、より少ないCPU時間を用いて更新データセットをより効率的に生成することを可能にするという利点を有する。例えば、コンピュータシステムは、各データセットを介した1度のみの実行により、好ましい第1のデータセット及び第2のデータセットの比較を可能にする。

【0018】

更に、コンピュータシステムは、更新データセットを含む更新メッセージを生成及び送信するセレクタに関連するコミュニケーションを含む。メッセージは、リモート端末に向けて送信すべきデータセットを含むリスト、配列、ビットボード、スタック、ヒープ、木又はコレクション、その他等のデータ構成を含む。好ましくは、メッセージはFIX基準を用いて送信されるが、Omneet API、XTP、SSL又は他の任意の基準等の当業者がメッセージ送信に使用する周知の任意の他のプロトコル、又は独自仕様のプロトコルであってもよい。

【0019】

好ましくは、第1のデータセット及び第2のデータセットは、時間とともに変化する動的データ要素を含む。例えば、第1のデータセットは、時刻T1での商取引システムにおけるオーダー・ブック(order book)であり、第2のデータセットは、時刻T2でのオーダー・ブックであってもよい。販売金融商品に関する購入又は売り出しといった新たなオーダーが入力される可能性があるため、オーダー・ブック内のデータは、時刻T1とT2との間で変更される可能性がある。情報配布システムは、ある所定時間間隔か又はオーダー・ブック内のアクティブ時かのいずれかにおいて、リモート端末に向けて情報を配布してもよい。例えば、オーダー・ブックが新たなオーダーを全く受信しなければ、リモート端末に向けて新たな更新を送信することは意味がない。しかし、新たなオーダーがオーダー・ブックに入力された場合、新たな情報は、リモート端末に伝播される必要がある。更に、アクティブが発生した時点において、更新メッセージを送信する必要がある。他の例では、オーダー・ブック内で生じる変更が状態遷移している間、例えば、オーダー・ブックが開く又閉じられたときである。

【0020】

従って、第2のデータセットは、第1のデータセットよりも新しいバージョンである。このように中央システムは、生じた変更を比較でき、リモート端末の有するデータセット

10

20

30

40

50

を把握できるので、格納及び送信するための演算子を更新データセット内から抽出できる。また更に、リモート端末側のユーザがデータセットの異なる部分に関する情報の取得を所望する場合があるので、第1のデータセット及び第2のデータセットの間の変更の確実な部分を特定のリモート端末に向けて送信することを可能にする。

【0021】

好ましくは、演算子は、追加演算子(Add operator)、削除演算子(Delete operator)、置換演算子(Replace operator)を含むグループから決められる。変更パラメータに基づいてこれら演算子を決定することにより、更新データセットの生成が可能になる。演算子は、本明細書で後述する組み合わせがなされてもよい。

【0022】

変更パラメータは、好ましくは、第1のデータセットに関連する第1のカウンタと、第2のデータセットに関連する第2のカウンタとを含む。このようにして選択処理は、より正確な方法でモニタ及び管理されうる。更に、セレクタは、第1のデータセットに関連する第1のカウンタと、第2のデータセットに関連する第2のカウンタとの関係に基づいて演算子を決定する。その関係は、>、<、=、 \neq 又は \in を含む関係群の中から選択された関係であるため、これにより、選択処理が高速化する。カウンタ間の関係に基づいて、追加(Add)、削除(Delete)、置換(Replace)等の演算子の少なくとも1つに決められる。

【0023】

更に、変更パラメータは、第1のデータセット及び第2のデータセット内におけるコンパレータの論理位置の追跡を維持するために、第1のデータセットに関連する第1の位置パラメータと、第2のデータセットに関連する第2の位置パラメータとを含む。このように第1のリスト及び第2のリストを順次比較した結果は、より正確な方法でモニタ及び管理されうる。

【0024】

演算子は、差分変更を含む。従って、データ要素の一部のみが変更したのであれば、変更したデータ要素の当該部分を記述する演算子であってもよい。しかし、演算子は、例えば、削除、置換又は追加されるべきデータ要素全てを記述できてもよい。

【0025】

各データ要素は、通常、少なくともキーを含む。しかし、データ要素は、データ部分を含んでもよい。一般に、データ部分は、空であってもよい。しかし、商取引システムにおいて、一般的なキーは、オーダー・ブック内における値段である。データ部分は、例えば、その値段の合計額であってもよい。しかし、キーが値段及び時間(time)を含むのであれば、全てのキーを送信することは必ずしも必要ではない。要素のソートは、値段及び時間に基づいて行なわれる場合もあるが、リモート端末に向けて送信される値段のみに基づいて行なわれてもよい。

【0026】

本発明の第2の側面においては、電子商取引システムにより上記及び他の目的を解決する。電子商取引システムは、上述したコンピュータシステムを含む。

【0027】

コンピュータシステムは、電子商取引システム内における内蔵モジュールであってもよい。また、情報抽出システムとして別々に販売されうるスタンドアローンモジュールであってよい。

【0028】

本発明の第3の側面においては、方法により上記及び他の目的を解決する実現する。方法は、リモート端末に向けて送信すべき更新データセットを生成する。更新データセットは、ソート済みのデータ要素を含む第1のデータセットとソート済みのデータ要素を含む第2のデータセットとの差分を記述する演算子を含む。方法は、

第1のデータセット内のデータ要素と第2のデータセット内のデータ要素とを順次比較するステップであって、先の比較結果が、第1のデータセット内のどのデータ要素と第2

10

20

30

40

50

のデータセット内のどのデータ要素が次の比較において比較されるかを制御するステップと、

各比較の後にメモリ内の変更パラメータを更新するステップと、

第2のデータセット内のデータ要素が第1のデータセット内のデータ要素に一致することを検出すると、メモリに格納された変更パラメータに基づいて演算子を決定する選択処理を開始するステップと、

決定された演算子をメモリに格納するステップと

を含み、

第1のデータセット及び第2のデータセットの差分を記述する演算子を含むリモート端末に送信すべき更新データセットを生成する。

10

【0029】

方法は、商取引システム等のコンピュータシステムが、より少ないCPU時間を用いて更新データセットをより効率的に生成することを可能にするという利点を有する。本発明は、例えば、各データセットを介した1度のみの実行により、好ましい第1のデータセット及び第2のデータセットの比較を可能にする。

【0030】

更に、方法は、決定された演算子と比較済みのデータ要素とを関連付けるステップを含む。このようにシステムは、演算子が使用されるべきデータ要素での追跡の維持を可能にする。

【0031】

また、方法は、追加演算子、削除演算子、置換演算子を含むグループから少なくとも1つの演算子を決定するステップを含む。変更パラメータに基づいてこれら演算子を決定することにより、更新データセットの生成が可能になる。演算子は、本明細書で後述する組み合わせがなされてもよい。

20

【0032】

上述した変更パラメータは、第1のデータセットに関連する第1のカウンタと、第2のデータセットに関連する第2のカウンタとを含んでもよい。従って、方法は、第1のデータセットに関連する第1のカウンタと、第2のデータセットに関連する第2のカウンタとの関係に基づいて演算子を決定するステップを更に含んでもよい。

【0033】

30

好ましくは、各データ要素は、キー及びデータ部分を含む。方法は、第1のデータセットの要素のキー及びデータ部分の少なくとも1つと、第2のデータセットの要素のキー及びデータ部分の少なくとも1つとを比較するステップを更に含む。

【0034】

上述した第1のデータセット及び第2のデータセットは、時間とともに変化する動的情報を含んでもよい。

【0035】

本発明の第4の側面においては、コンピュータ・プログラムにより上記及び他の目的を解決する。コンピュータ・プログラムは、上述した側面及び実施形態の少なくとも1つに係わり、コンピュータ可読媒体に格納される。

40

【0036】

本発明のこれら及び他の側面は、後述する実施形態を参照して明白及び明確に示されるであろう。

【図面の簡単な説明】

【0037】

【図1】本発明で用いられるコンピュータネットワークの概要の一例を示す図である。

【図2】メモリ、第1のデータセット、第2のデータセット、変更パラメータ、コンパレータ、セレクタ、更新データセット、インターフェースを含む電子装置を示す図である。

【図3】データ要素を含むデータセットとキー及びデータを含むデータ要素とを示す図である。

50

- 【図4】テキストに記載されるデータセットを反復するアルゴリズムを示す図である。
- 【図5】データセットA及びデータセットBを示す図である。
- 【図6】データセットAの要素とデータセットBの要素との比較の一例を示す図である。
- 【図7】データセットAの要素とデータセットBの要素との比較の一例を示す図である。
- 【図8】データセットAの要素とデータセットBの要素との比較の一例を示す図である。
- 【図9】データセットAの要素とデータセットBの要素との比較の一例を示す図である。
- 【図10】データセットAの要素とデータセットBの要素との比較の一例を示す図である。
- 。
- 【図11】選択方法により選択された第1の演算子を示す図である。
- 【図12】データセットAの要素とデータセットBの要素との比較の一例を示す図である 10
- 。
- 【図13】データセットAの要素とデータセットBの要素との比較の一例を示す図である
- 。
- 【図14】データセットAの要素とデータセットBの要素との比較の一例を示す図である
- 。
- 【図15】データセットAの要素とデータセットBの要素との比較の一例を示す図である
- 。
- 【図16】データセットAの要素とデータセットBの要素との比較の一例を示す図である
- 。
- 【図17】データセットAの要素とデータセットBの要素との比較の一例を示す図である 20
- 。
- 【図18】選択方法により選択された更なる演算子を示す図である。
- 【図19】選択方法により選択された演算子を含む更新データセットの一例を示す図である。
- 【図20】演算子が使用されるより以前のデータセットA及びデータセットBを示す図である。
- 【図21】更新データセットにおける列の一例を示す図である。
- 【図22】第1の演算子を適用した後のデータセットAの変更を示す図である。
- 【図23】更新データセットにおける列の一例を示す図である。
- 【図24】第2の演算子を適用した後のデータセットAの変更を示す図である 30
- 【図25】更新データセットにおける列の一例を示す図である。
- 【図26】第3の演算子を適用した後のデータセットAの変更を示す図である。
- 【図27】更新データセットにおける列の一例を示す図である。
- 【図28】第4の演算子を適用した後のデータセットAの変更を示す図である。
- 【図29】演算子がデータセットAに適用された後の結果を示す図である。
- 【図30】選択方法Bを示す図である。
- 【発明を実施するための形態】
- 【0038】
- 図1は、リモート端末1と、中央コンピュータ2と、ゲートウェイ又はルータ3とを含むコンピュータネットワークを示す。各装置間は、種々の太さを有するライン4により示される接続がなされている。ラインの太さは、帯域幅（データレート）を示す。太いラインは高いデータレートを有し、細いラインは低いデータレートを有している。図1に示す3つのフロントエンドコンピュータは、画面上に文字を有しており、この文字は、ユーザA、ユーザB、ユーザCに属するリモート端末であることを示している。更新データセット6は、ライン4と、ルータや外部等のネットワーク装置とを含むネットワークを介して、中央コンピュータ2からリモート端末1に向けて送信される。更新データセット6は、本発明に係わるコンピュータシステム5において生成される。 40
- 【0039】
- 図2は、本発明に係わるコンピュータシステム5を示す。コンピュータシステム5は、メモリ10と、コンパレータ14と、セレクタ15と、更新データセット7を送信するた 50

めのインターフェース 16とを含む。メモリ 10は、第1のデータセット 11及び第2のデータセット 12を含む。更に、メモリは、変更パラメータ 13と、更新データセット 7と、演算子 18とを格納するための領域を含む。

【0040】

図3は、データセット 11と、データセットにおけるデータ要素 17の拡大とを示す。図中データ要素 17は、キーとデータ部分との両方を含んでいるが、本発明の他の実施形態においては、データ要素 17は、キー部分を少なくとも含んでいればよい。

【0041】

以下、図を参照して本発明の詳細について説明する。演算子がどのようにして決定／選択されるかの詳細について説明する。

10

【0042】

以下、2つのデータセット間の差分を効率的に記述する演算子を選択するための方法の一例を示す。

【0043】

データセットは、1又は複数の要素から構成される。各要素は、キーと、可能であればデータ部分とを有する。キーは、データセット内の要素各自に論理位置を与えるソートアルゴリズムと対応付けられる。

【0044】

この方法は、1)及び2)に基づく：

1) 演算子の効率的なセットの策定

20

2) 2つのデータセットの差分を評価し、演算子のセットを用いて当該差分を記述する効率的なアルゴリズムの策定。これを実現するため、アルゴリズムは、2つのデータセットのワン・パス・トラバース (one-pass traverse) を使用する。

【0045】

この方法は、下記に示す2つのデータセット間における変更を記述する演算子のセットを使用する：

演算子	意味
追加(Add)	所定の論理位置に新たな要素を追加する
削除(Delete(N))	所定の論理位置から1又は複数(N)の要素を削除する
置換(Replace)	所定の論理位置の要素を置換する

30

【0046】

[方法紹介]

2つのデータセット内で一致する要素（キー及びデータ部分の両方）は変わらない。アルゴリズムは、変わらない要素をバリア (barrier) とみなす。バリアは、演算子の最適化が実行されうる時点を識別するために上記方法により使用される。当該方法は、監視に基づいており、バリアが検出される前に使用された演算子に関して最適化が可能になる。

40

【0047】

図5を参照してデータセット A 及びデータセット B を想定してみる。まず、データセット A では、複数の演算子がデータセット B を獲得するのに適用されうる。以下に示すアルゴリズム及び方法は、データセット A をデータセット B に変換するこれら演算子を抽出する。

【0048】

データセットのトラバースの間、好ましくは、以下に示すカウンタ、# D e l 及び # A

50

`d d` が使用される。これらには、最初に、`0` 又は同じ目的を果たす任意の値が設定される。

【 0 0 4 9 】

その後、データセット A 及びデータセット B は、キーのソート順番に従って、好ましくは開始からトラバースされる。

【 0 0 5 0 】

データセット A 内における実際の論理位置は、以降、`A p o s` と名付けられる。`A p o s` は、データセット A に関する位置パラメータを含む。データセット B 内における実際の論理位置は、以降、`B p o s` と名付けられる。`B p o s` は、データセット B に関する位置パラメータを含む。`A p o s` 及び `B p o s` は、最初に、データセット A 及びデータセット B 内における最初の論理位置に設定される。

10

【 0 0 5 1 】

初期化：

`A d d` 及び # `D e l` に対して `0` を設定する。

`A p o s` 及び `B p o s` に対してデータセット A 及びデータセット B 内における最初の論理位置を設定する。

`A p o s S t a r t` に対して `A p o s` を設定し、`B p o s S t a r t` に対して `B p o s` を設定する。

【 0 0 5 2 】

アルゴリズム：データセット A 及びデータセット B を反復し、# `A d d` 及び # `D e l` をカウントする：

20

1. 論理位置 `A p o s` 及び `B p o s` において、セット A の要素とセット B の要素とを比較する。

a . `A` 要素及び `B` 要素がなければ、以下に示す方法 B を用いて演算子を選択し、ステップ 2 を継続する（完了）。

b . キーは一致するがデータ部分が異なれば、# `A d d`、# `D e l`、`A p o s`、`B p o s` を 1 増加する。ステップ 1 を継続し、次の要素を比較する。

c . `A` 要素キーが `B` 要素キーよりも（ソート順に照らして）悪い、又は `A` 要素がなければ、# `A d d` 及び `B p o s` を 1 増加する。ステップ 1 を継続し、次の要素を比較する。

30

d . `A` 要素キーが `B` 要素キーよりも（ソート順に照らして）良い、又は `B` 要素がなければ、# `D e l` 及び `A p o s` を 1 増加する。ステップ 1 を継続し、次の要素を比較する。

e . キー及びデータ部分の両方が一致すれば、バリアが発見される。（# `A d d` = 0 又は # `D e l` = 0）であれば、以下に示す方法 B を用いて演算子を選択する。

継続：

`A d d` 及び # `D e l` に対して `0` を設定する。

`A p o s` 及び `B p o s` を 1 増加する。

`A p o s S t a r t` に `A p o s`、`B p o s S t a r t` に `B p o s` を設定する。

ステップ 1 を継続し、次の要素を比較する。

【 0 0 5 3 】

2 . 完了

40

方法 B : 演算子の選択 : # `D e l` と # `A d d` とを比較。

a . # `D e l` が # `A d d` を越えていれば、以下の演算子を選択する。

論理位置 `A p o s S t a r t` において、($N = \# D e l$) 演算子を削除する。

続いて以下を行なう：

論理位置 `A p o s S t a r t` から開始し、セット B 要素から # `A d d` 数の演算子を追加する。

b . # `D e l` と # `A d d` とが等しければ、以下の演算子を選択する。

論理位置 `A p o s S t a r t` から開始し、セット B 要素から # `A d d` 数の演算子を置換する。

c . # `D e l` が # `A d d` よりも小さければ、以下の演算子を選択する。

50

`A p o s S t a r t + # D e l` から開始し、セットB要素から (`# A d d - # D e l`) 数の演算子を追加する。

【0054】

[例]

以下、図を参照して種々の方法ステップの一例について説明する。

図5に示すテーブルに従って、データセットA及びデータセットBを想定する。要素は、好ましくは、キーに基づいてソートされ、その結果、論理位置1において最も高い値が得られる。データセットAは古いデータセットを示し、データセットBは新しいデータセットを示す。このタスクは、データセットAをデータセットBに変換することである。

【0055】

[方法の実行]

初期化：

`# A d d` 及び `# D e l` に対して0を設定する。

`A p o s` 及び `B p o s` に対してデータセットA及びデータセットB内における最初の論理位置を設定する。

`A p o s` 及び `B p o s` に1を設定する。

`A p o s S t a r t` に `A p o s` を設定し、`B p o s S t a r t` に `B p o s` を設定する。

`A p o s S t a r t` 及び `B p o s S t a r t` に1を設定する。

【0056】

方法及びシステムは、カウンタの初期化を開始する。カウンタは、好ましくは、データセットの反復に使用されるであろう。

【0057】

その後、反復フェーズが開始される。反復フェーズでは、変更パラメータ `# A d d` 及び `# D e l` が反復及び増加される。

【0058】

[ステップ1]

セットAの要素とセットBの要素とを比較する。`A p o s` 及び `B p o s` には、両方1が設定されるので、これは、リストにおける1番目の要素が比較されることを意味する。しかし、データセットにおける要素が他の方法でソートされていれば、この方法では、まず、他の要素の比較を開始する場合もある。図6にこの比較を示す。この場合、99は100よりも悪い(アルゴリズムステップc参照)(データセットAの要素(`A p o s = 1`)は、データセットBの要素(`B p o s = 1`)よりも悪い)。そのため、`# A d d`を1増加し、`# A d d`に1を設定する。`B p o s`を1増加し、`B p o s`に2を設定する。アルゴリズムにおけるステップ1を継続し、次の要素を比較する。

【0059】

[ステップ2]

セットAの要素とセットBの要素とを比較する。ここでは、`B p o s`カウンタが1増加しているため、データセットAの(`A p o s = 1`)における要素と、データセットBの(`B p o s = 2`)における要素とが比較される。この比較を図7に示す。99は100よりも悪い(アルゴリズムステップc参照)。そのため、`# A d d`を1増加し、`# A d d`に2を設定する。`B p o s`を1増加し、`B p o s`に3を設定する。アルゴリズムにおけるステップ1を継続し、次の要素を比較する。

【0060】

[ステップ3]

上記同様に、セットAの要素とセットBの要素とを比較する。この比較を図8に示す。99は97よりも良い(アルゴリズムステップd参照)。そのため、`# D e l`を1増加し、`# D e l`に1を設定する。`A p o s`を1増加し、`A p o s`に2を設定する。アルゴリズムにおけるステップ1を継続し、次の要素を比較する。

【0061】

[ステップ4]

10

20

30

40

50

セットAの要素とセットBの要素とを比較する。この比較を図9に示す。98は97よりも良い(アルゴリズムステップd参照)。そのため、#De1を1増加し、#De1に2を設定する。Aposを1増加し、Aposに3を設定する。アルゴリズムにおけるステップ1を継続し、次の要素を比較する。

【0062】

[ステップ5]

セットAの要素とセットBの要素とを比較する。この比較を図10に示す。キー及びデータ部分の両方が一致するため(アルゴリズムステップe参照)、バリアに到達している。セレクタを起動し、方法Bを用いて演算子を選択する。

【0063】

[ステップ6]

#De1と#Addとを比較する。方法Bによれば、ステップbが、この状況:(b)に適用される。#De1(2) = #Add(2)であるので、下記に示す演算子を選択する:

置換(Replace)2(#Add)演算子、セットBからデータを選択し、論理位置AposStart = 1から開始する。選択された演算子を図11に示す。

【0064】

[ステップ7]

データセットA及びデータセットBにおける要素の反復及び比較は継続する。

#Add及び#De1に対して0を設定する。

バリアを通り抜けるため、カウンタが増加される。Apos及びBposが1増加され、Apos及びBposに4が設定される。

AposStartにApos、BposStartにBposを設定し、AposStart及びBposStartに4を設定する。

アルゴリズムにおけるステップ1を継続し、次の要素を比較する。

【0065】

[ステップ8]

セットAの要素とセットBの要素とを比較する。この比較を図12に示す。96は93よりも良い(アルゴリズムステップd参照)。そのため、#De1に1増加し、#De1に1を設定する。Aposを1増加し、Aposに5を設定する。アルゴリズムにおけるステップ1を継続し、次の要素を比較する。

【0066】

[ステップ9]

セットAの要素とセットBの要素とを比較する。この比較を図13に示す。95は93よりも良い(アルゴリズムステップd参照)。そのため、#De1に1増加し、#De1に2を設定する。Aposを1増加し、Aposに6を設定する。アルゴリズムにおけるステップ1を継続し、次の要素を比較する。

【0067】

[ステップ10]

セットAの要素とセットBの要素とを比較する。この比較を図14に示す。94は93よりも良い(アルゴリズムステップd参照)。そのため、#De1に1増加し、#De1に3を設定する。Aposを1増加し、Aposに7を設定する。アルゴリズムにおけるステップ1を継続し、次の要素を比較する。

【0068】

[ステップ11]

セットAの要素とセットBの要素とを比較する。この比較を図15に示す。キーは一致するが、データ部分が異なる(10は、15と等しくない)(アルゴリズムステップb参照)。

#Addを1増加し、#Addに1を設定する。

#De1を1増加し、#De1に4を設定する。

10

20

30

40

50

A pos を 1 増加し、A pos に 8 を設定する。

B pos を 1 増加し、B pos に 5 を設定する。

アルゴリズムにおけるステップ 1 を継続し、次の要素を比較する。

【0069】

[ステップ12]

セット A の要素とセット B の要素とを比較する。この比較を図 16 に示す。B 要素がない（空である）（アルゴリズムステップ d 参照）。

De1 を 1 増加し、# De1 に 5 を設定する。A pos を 1 増加し、A pos に 9 を設定する。アルゴリズムにおけるステップ 1 を継続し、次の要素を比較する。

【0070】

10

[ステップ13]

セット A の要素とセット B の要素とを比較する。この比較を図 17 に示す。A pos (9) におけるデータセット A の要素と、B pos (5) におけるデータセット B の要素とが空となる。そのため、アルゴリズムのステップ a が適用される。アルゴリズムにおけるステップ a は、セレクタを起動する。ステップ 14 に示されている方法 B である。

上述した選択の後、ステップ 2 を継続する（完了）。

【0071】

[ステップ14]

選択方法 B を使用することにより、カウンタ # De1 とカウンタ # Add とを比較する。

20

選択方法 B における選択ステップ (a) によれば、# De1 (5) が # Add (1) を越えているため、演算子は、好ましくは、下記に示す通りに選択される：

論理位置 A pos Start (4) において、5 (# De1) 演算子を削除する。

続いて以下を行なう：

論理位置 A pos Start (4) から開始し、セット B 要素から 1 (# Add) 演算子を追加する。

この演算子の選択を図 18 に示す。

【0072】

[ステップ15]

アルゴリズムにおける最後のステップ 2（完了）が上述したステップ 14 に到達する。

30

アルゴリズム及び方法 B の反復の完了後、データセット A からデータセット B への変更を記述する複数の完全な演算子が、好ましくは、対応する論理位置、キー、データ部分とともにメモリに格納される。図 19 に示すように、演算子、論理位置、キー、データ部分を含む更新データセットが生成される。

【0073】

他の実施形態においては、論理位置は、キーにより記述されうる。この場合、端末が受信する更新データセットが、好ましくは、データ要素のソートが行なえるように、相互に関連する論理位置の状態を示す情報を含む。これにより、論理位置が、1、2、3・・・等である必要はない。また、ソートを容易にするため、位置が相互に関連を有する限り、A、B、C・・・やその他により記述されうる。

40

【0074】

データセット A を承知した上で、更新データセットは、図 19 に示すような演算子を含み、最初から最後まで演算子を一つずつ順番に用いることにより、データセット B が生成されうる。

【0075】

これがどのようにしてなされるかの段階的な記載が添付の図 20～図 29 に示される。

【0076】

図 20 は、更新データセットがデータセット A に適用されるよりも以前のデータセット A 及びデータセット B の状況を示す。そのため、以下のステップは、図 19 における更新データセットを使用することにより、データセット A からデータセット B を生成する。

50

【 0 0 7 7 】

図21に示すように、図19に示す更新データセットの1番目の列をデータセットAに適用することにより、図22に示すように、修正されたデータセットAが得られる。

【 0 0 7 8 】

図23に示すように、図19に示す更新データセットの2番目の列を、データセットAに適用することにより、図24に示すように、修正されたデータセットAが得られる。

【 0 0 7 9 】

図25に示すように、図19に示す更新データセットの3番目の列を、データセットAに適用することにより、図26に示すように、修正されたデータセットAが得られる。

【 0 0 8 0 】

図27に示すように、図19に示す更新データセットの4番目の列を、データセットAに適用することにより、図28に示すように、修正されたデータセットAが得られる。

【 0 0 8 1 】

ここで、演算子は、図29に示すように、データセットAをデータセットBに変換する。

【 0 0 8 2 】

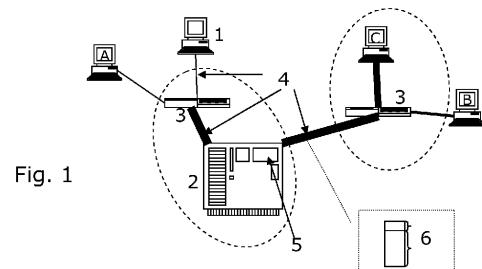
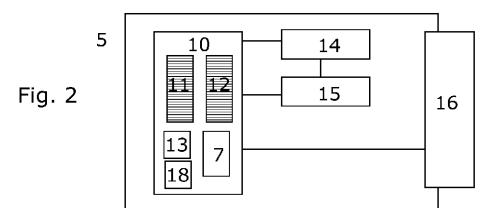
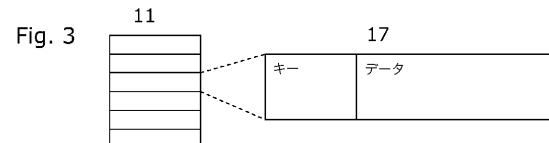
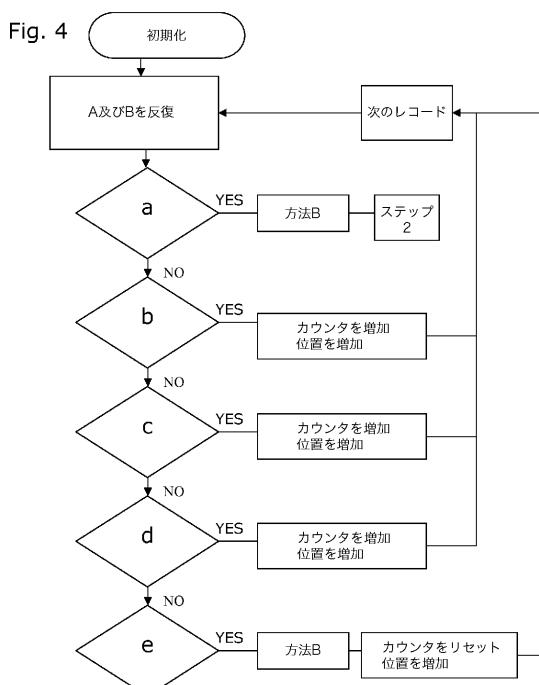
上記記載における用語”含む (c o m p r i s i n g) ”は、他の要素又はステップを除外しない。”単数 (a ”又は” a n ”) は、複数を除外しない。

【 0 0 8 3 】

更に、用語”含む (i n c l u d e) ”及び”含む (c o n t a i n) ”は、他の要素又はステップを除外しない。

10

20

【 図 1 】**【 図 2 】****【 図 3 】****【 図 4 】**

【図5】

FIG 5

データセットA		
論理位置	キー	データ部分
1	99	10
2	98	10
3	97	10
4	96	10
5	95	10
6	94	10
7	93	10
8	92	10

データセットB		
論理位置	キー	データ部分
1	101	10
2	100	10
3	97	10
4	93	15

【図9】

FIG 9

データセットA		
論理位置	キー	データ部分
2	98	10

データセットB		
論理位置	キー	データ部分
3	97	10

【図6】

FIG 6

データセットA		
論理位置	キー	データ部分
1	99	10

データセットB		
論理位置	キー	データ部分
1	101	10

【図7】

FIG 7

データセットA		
論理位置	キー	データ部分
1	99	10

データセットB		
論理位置	キー	データ部分
2	100	10

【図8】

FIG 8

データセットA		
論理位置	キー	データ部分
1	99	10

データセットB		
論理位置	キー	データ部分
3	97	10

【図14】

FIG 14

データセットA		
論理位置	キー	データ部分
6	94	10

データセットB		
論理位置	キー	データ部分
4	93	15

【図15】

FIG 15

データセットA		
論理位置	キー	データ部分
7	93	10

データセットB		
論理位置	キー	データ部分
4	93	15

【図16】

FIG 16

データセットA		
論理位置	キー	データ部分
8	92	10

データセットB		
論理位置	キー	データ部分
5		

【図17】

FIG 17

データセットA		
論理位置	キー	データ部分
9		

データセットB		
論理位置	キー	データ部分
5		

【図18】

FIG 18

演算子	論理位置	キー	データ部分
削除(5)	4		
追加	4	93	15

【図9】

FIG 9

データセットA		
論理位置	キー	データ部分
2	98	10

データセットB		
論理位置	キー	データ部分
3	97	10

【図10】

FIG 10

データセットA		
論理位置	キー	データ部分
3	97	10

データセットB		
論理位置	キー	データ部分
3	97	10

【図11】

FIG 11

演算子	論理位置	キー	データ部分
置換	1	101	10
置換	2	100	10

【図12】

FIG 12

データセットA		
論理位置	キー	データ部分
4	96	10

データセットB		
論理位置	キー	データ部分
4	93	15

【図19】

FIG 19

演算子	論理位置	キー	データ部分
置換	1	101	10
置換	2	100	10
削除(5)	4		

【図20】

FIG 20

データセットA		
論理位置	キー	データ部分
1	99	10
2	98	10
3	97	10
4	96	10
5	95	10
6	94	10
7	93	10
8	92	10

データセットB		
論理位置	キー	データ部分
1	101	10
2	100	10
3	97	10
4	93	15

【図21】

FIG 21

演算子	論理位置	キー	データ部分
置換	1	101	10

【図22】

FIG 22

データセットA

論理位置	キー	データ部分
1	99	10
	101	10
2	98	10
3	97	10
4	96	10
5	95	10
6	94	10
7	93	10
8	92	10

【図23】

FIG 23

演算子	論理位置	キー	データ部分
置換	2	100	10

【図24】

FIG 24

データセットA

論理位置	キー	データ部分
1	99	10
	101	10
2	98	10
	100	10
3	97	10
4	96	10
5	95	10
6	94	10
7	93	10
8	92	10

【図25】

FIG 25

演算子	論理位置	キー	データ部分
削除 (5)	4		

【図26】

FIG 26

データセットA

論理位置	キー	データ部分
1	99	10
	101	10
2	98	10
	100	10
3	97	10
4	96	10
5	95	10
6	94	10
7	93	10
8	92	10

【図28】

FIG 28

データセットA

論理位置	キー	データ部分
1	99	10
	101	10
2	98	10
	100	10
3	97	10
4	96	10
	93	15
5	95	10
6	94	10
7	93	10
8	92	10

【図27】

FIG 27

演算子	論理位置	キー	データ部分
追加	4	93	15

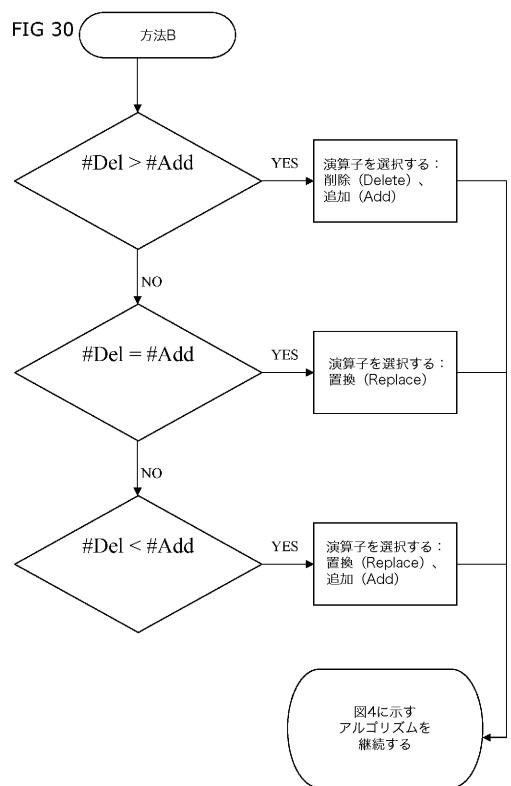
【図29】

FIG 29

データセットB

論理位置	キー	データ部分
1	101	10
2	100	10
3	97	10
4	93	15

【図30】



フロントページの続き

(74)代理人 100115071

弁理士 大塚 康弘

(74)代理人 100116894

弁理士 木村 秀二

(74)代理人 100130409

弁理士 下山 治

(72)発明者 ランツ， スタファン

スウェーデン国 ストックホルム エス - 115 30 , スケッパルガタン 100

(72)発明者 ヤンソン， ラース

スウェーデン国 ブロンマ エス - 168 56 , ベッコムベルガヴェーゲン 89

審査官 田川 泰宏

(56)参考文献 Wilbert Labio , Efficient snapshot differential algorithms for data warehousing , Proceeding VLDB '96 Proceedings of the 22th International Conference on Very Large Data , 米国 , Morgan Kaufmann Publishers Inc. , 1996年 , p.63 - 74 , U R L , <http://www.vldb.org/conf/1996/P063.PDF>

(58)調査した分野(Int.Cl. , DB名)

G 06 F 12 / 00

G 06 F 13 / 00

G 06 F 17 / 30