

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2012-155634
(P2012-155634A)

(43) 公開日 平成24年8月16日(2012.8.16)

(51) Int.Cl. F I テーマコード(参考)
G06F 12/00 (2006.01) G06F 12/00 510B 5B082
 G06F 12/00 531J

審査請求 未請求 請求項の数 5 O L (全 19 頁)

(21) 出願番号 特願2011-15894(P2011-15894)
 (22) 出願日 平成23年1月28日(2011.1.28)

(71) 出願人 000237639
 富士通フロンテック株式会社
 東京都稲城市矢野口1776番地
 (74) 代理人 100092152
 弁理士 服部 毅巖
 (72) 発明者 螺良 修一
 東京都稲城市矢野口1776番地 富士通
 フロンテック株式会社内
 Fターム(参考) 5B082 DD08 GA04

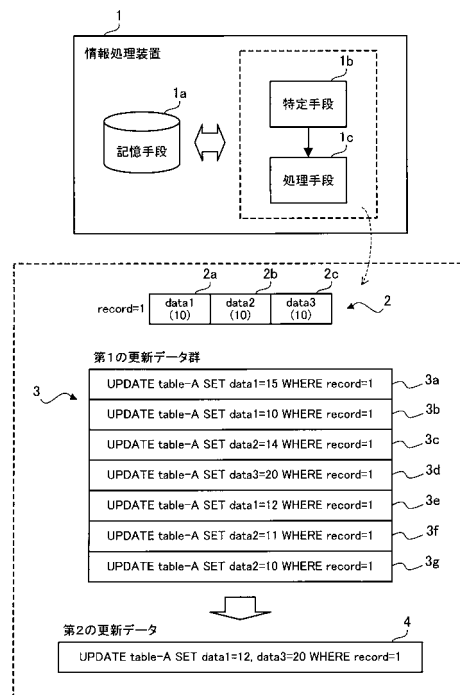
(54) 【発明の名称】 情報処理プログラム、情報処理装置および情報処理方法

(57) 【要約】

【課題】更新ログのデータ量を効率的に低減すること。

【解決手段】記憶手段 1 a は、データ項目 2 a , 2 b , 2 c を含むレコード 2 の初期設定と、データ項目 2 a , 2 b , 2 c に対する初期設定からの更新内容を示す第 1 の更新データ群 3 とを記憶する。特定手段 1 b は、記憶手段 1 a を参照して、データ項目 2 a , 2 b , 2 c のうち、最新の更新データによる設定が初期設定と同一でないデータ項目 2 a , 2 c につき最新の更新データ 3 d , 3 e を特定する。処理手段 1 c は、特定手段 1 b により特定された更新データ 3 d , 3 e を用いて、データ項目 2 a , 2 b , 2 c に対する 1 つの第 2 の更新データ 4 を生成して記憶手段 1 a に格納し、記憶手段 1 a から第 1 の更新データ群 3 を削除する。

【選択図】 図 1



【特許請求の範囲】**【請求項 1】**

コンピュータを、

複数のデータ項目を含むレコードの初期設定と、前記複数のデータ項目に対する前記初期設定からの更新内容を示す複数の第 1 の更新データとを記憶する記憶手段、

前記記憶手段を参照して、前記複数のデータ項目それぞれのうち、最新の更新データによる設定が前記初期設定と同一でないデータ項目につき前記最新の更新データを特定する特定手段、

前記特定手段により特定された全てのデータ項目の前記最新の更新データを用いて、前記複数のデータ項目に対する 1 つの第 2 の更新データを生成して前記記憶手段に格納し、
前記記憶手段から前記複数の第 1 の更新データを削除する処理手段、

として機能させることを特徴とする情報処理プログラム。

10

【請求項 2】

前記処理手段は、前記特定手段により前記複数のデータ項目の何れかにつき前記最新の更新データとして前記レコードの削除が特定された場合、前記レコードを削除するための処理の記述を含む前記第 2 の更新データを生成する、

ことを特徴とする請求項 1 記載の情報処理プログラム。

【請求項 3】

前記コンピュータを、

前記記憶手段に記憶された前記第 2 の更新データを用いて前記レコードの設定と他のコンピュータが記憶するレコードの設定とを同期させる同期処理手段、

として機能させることを特徴とする請求項 1 または 2 の何れか一項に記載の情報処理プログラム。

20

【請求項 4】

複数のデータ項目を含むレコードの初期設定と、前記複数のデータ項目に対する前記初期設定からの更新内容を示す複数の第 1 の更新データとを記憶する記憶手段と、

前記記憶手段を参照して、前記複数のデータ項目それぞれのうち、最新の更新データによる設定が前記初期設定と同一でないデータ項目につき前記最新の更新データを特定する特定手段と、

前記特定手段により特定された全てのデータ項目の前記最新の更新データを用いて、前記複数のデータ項目に対する 1 つの第 2 の更新データを生成して前記記憶手段に格納し、
前記記憶手段から前記複数の第 1 の更新データを削除する処理手段と、

を有することを特徴とする情報処理装置。

30

【請求項 5】

コンピュータの情報処理方法であって、

複数のデータ項目を含むレコードの初期設定と、前記複数のデータ項目に対する前記初期設定からの更新内容を示す複数の第 1 の更新データとを記憶する記憶手段を参照して、前記複数のデータ項目それぞれのうち、最新の更新データによる設定が前記初期設定と同一でないデータ項目につき前記最新の更新データを特定し、

特定した全てのデータ項目の前記最新の更新データを用いて、前記複数のデータ項目に対する 1 つの第 2 の更新データを生成して前記記憶手段に格納し、前記記憶手段から前記複数の第 1 の更新データを削除する、

ことを特徴とする情報処理方法。

40

【発明の詳細な説明】**【技術分野】****【0001】**

本発明は更新ログを処理する情報処理プログラム、情報処理装置および情報処理方法に関する。

【背景技術】**【0002】**

50

従来、システムとしての可用性や信頼性を向上するため、構成部品を冗長化して運用する情報処理システムが利用されている。その1つとして、例えば、データベースを冗長化するものがある。具体的には、一方のデータベースを運用系として業務処理に用い、他方のデータベースを待機系として用いる。当該情報処理システムは運用系での更新内容を待機系にも反映して、両方のデータベースを同期する。そして、運用系の障害時には、運用系での業務処理を待機系に引き継がせる。

【0003】

ここで、データベースを冗長化する場合、両方のデータベースでデータの同期を高速に行えることが、信頼性の観点からは望ましい。そこで、一方のデータベースでの差分を記録した差分ログを取得し、当該差分ログを複数の並列処理対象に振り分けて、当該並列処理対象の差分ログを他方のデータベースに適用する処理を並列に行うことで、差分ログを高速に処理する方法が提案されている（例えば、特許文献1参照）。

10

【0004】

また、データベースの同じデータにアクセスする可能性のあるトランザクション同士を同じコネクションにまとめ、コネクションごとの複数のトランザクションを一まとめにしてコミットを実行することで、トランザクションの速度を高速化する方法も提案されている（例えば、特許文献2参照）。

【先行技術文献】

【特許文献】

【0005】

20

【特許文献1】国際公開第2004/090726号

【特許文献2】特開2009-26029号公報

【発明の概要】

【発明が解決しようとする課題】

【0006】

ところで、障害の起きたデータベースが復旧した場合に、当該復旧したデータベースにそれまでの更新内容を反映することが考えられる。そのために、運用中のもう一方のデータベースで、更新内容を記録した更新ログを取得することがある。

【0007】

その場合、片方のデータベースのみによる運用時間が長いほど、更新ログに蓄積されるデータ量は増加する。このため、一方が障害から復旧して、差分を同期しようとする場合に、更新ログのデータ量の分だけ処理時間が長くなるという問題がある。また、更新ログの最大データ量が所定量に制限されていることもあり、当該所定量を超えた更新ログの取得が困難になるという問題がある。

30

【0008】

本発明はこのような点に鑑みてなされたものであり、更新ログのデータ量を効率的に低減できる情報処理プログラム、情報処理装置および情報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】

40

コンピュータを記憶手段、特定手段および処理手段として機能させる情報処理プログラムが提供される。記憶手段は、複数のデータ項目を含むレコードの初期設定と、複数のデータ項目に対する初期設定からの更新内容を示す複数の第1の更新データとを記憶する。特定手段は、記憶手段を参照して、複数のデータ項目それぞれのうち、最新の更新データによる設定が初期設定と同一でないデータ項目につき最新の更新データを特定する。処理手段は、特定手段により特定された全てのデータ項目の最新の更新データを用いて、複数のデータ項目に対する1つの第2の更新データを生成して記憶手段に格納し、記憶手段から複数の第1の更新データを削除する。

【0010】

また、情報処理プログラムを実行するコンピュータと同様の機能を有する情報処理装置

50

が提供される。

また、コンピュータの情報処理方法が提供される。この情報処理方法では、コンピュータが、複数のデータ項目を含むレコードの初期設定と、複数のデータ項目に対する初期設定からの更新内容を示す複数の第1の更新データとを記憶する記憶手段を参照して、複数のデータ項目それぞれのうち、最新の更新データによる設定が初期設定と同一でないデータ項目につき最新の更新データを特定する。そして、特定した全てのデータ項目の最新の更新データを用いて、複数のデータ項目に対する1つの第2の更新データを生成して記憶手段に格納し、記憶手段から複数の第1の更新データを削除する。

【発明の効果】

【0011】

上記情報処理プログラム、情報処理装置および情報処理方法によれば、更新ログのデータ量を効率的に低減できる。

【図面の簡単な説明】

【0012】

【図1】第1の実施の形態の情報処理装置を示す図である。

【図2】第2の実施の形態の金融窓口システムを示す図である。

【図3】各装置の関係を示す図である。

【図4】振込ワークフローを例示する図である。

【図5】DBサーバのハードウェア構成を示す図である。

【図6】DBサーバの機能構成を示す図である。

【図7】伝票管理テーブルのデータ構造例を示す図である。

【図8】更新ログファイルのデータ構造例を示す図である。

【図9】最新レコードテーブルのデータ構造例を示す第1の図である。

【図10】最新レコードテーブルのデータ構造例を示す第2の図である。

【図11】更新ログ圧縮処理を示すフローチャートである。

【図12】圧縮処理後の更新ログファイルを例示する第1の図である。

【図13】圧縮処理後の更新ログファイルを例示する第2の図である。

【図14】圧縮処理後の更新ログファイルを例示する第3の図である。

【発明を実施するための形態】

【0013】

以下、本実施の形態を図面を参照して説明する。

[第1の実施の形態]

図1は、第1の実施の形態の情報処理装置を示す図である。情報処理装置1は、データベース上のテーブル“table-A”に関する更新ログを管理する。更新ログは、冗長化されたデータベースにつき両方の同期を行うためのものである。情報処理装置1は、記憶手段1a、特定手段1bおよび処理手段1cを有する。

【0014】

記憶手段1aは、レコード2の初期設定を記憶する。初期設定とは、更新ログの取得が開始された時点におけるレコード2の設定内容を示す。複数のレコードがある場合、初期設定はレコードごとに記憶される。当該レコードに複数のデータ項目があれば、初期設定は更にデータ項目ごとに記憶される。

【0015】

レコード2は、複数のデータ項目を含む。例えば、データ項目2a, 2b, 2cを含むとする。ここで、データ項目2aは“data1”、データ項目2bは“data2”、データ項目2cは“data3”の項目名が付されている。初期設定は、データ項目2a, 2b, 2cの何れも“10”であるとする。

【0016】

記憶手段1aは、データ項目2a, 2b, 2cに対する初期設定からの更新ログとして、第1の更新データ群3を記憶する。第1の更新データ群3は、例えば時系列の古い順に更新データ3a, 3b, 3c, 3d, 3e, 3f, 3gを有する。

10

20

30

40

50

【 0 0 1 7 】

特定手段 1 b は、記憶手段 1 a を参照して、データ項目 2 a , 2 b , 2 c それぞれのうち、最新の更新データによる設定が初期設定と同一でないデータ項目につき最新の更新データを特定する。

【 0 0 1 8 】

例えば、データ項目 2 a に関する最新の更新データ 3 e は “ 1 2 ” への更新を示す。よって、初期設定 “ 1 0 ” と同一でないので、特定手段 1 b は当該最新の更新データ 3 e を特定する。また、データ項目 2 b に関する最新の更新データ 3 g は “ 1 0 ” への更新を示す。よって、初期設定 “ 1 0 ” と同一であるので、特定の対象外となる。また、データ項目 2 c に関する最新の更新データ 3 d は “ 2 0 ” への更新を示す。よって、初期設定 “ 1 0 ” と同一でないので、特定手段 1 b は当該最新の更新データ 3 d を特定する。

10

【 0 0 1 9 】

処理手段 1 c は、特定手段 1 b により特定された全てのデータ項目の最新の更新データを用いて、複数のデータ項目に対する 1 つの第 2 の更新データ 4 を生成して記憶手段 1 a に格納する。そして、処理手段 1 c は記憶手段 1 a から第 1 の更新データ群 3 を削除する。第 2 の更新データ 4 は、冗長化された両方のデータベースの同期を行うために、第 1 の更新データ群 3 に代えて用いることができる。

【 0 0 2 0 】

例えば、上述の例では特定手段 1 b は、データ項目 2 a に対して更新データ 3 e を、データ項目 2 c に対して更新データ 3 d を特定している。処理手段 1 c は、当該更新データ 3 d , 3 e を用いて、第 2 の更新データ 4 を生成する。第 2 の更新データ 4 は、データ項目 2 a を “ 1 2 ” に、データ項目 2 c を “ 2 0 ” に、それぞれ更新するためのデータである。そして、処理手段 1 c は記憶手段 1 a から第 1 の更新データ群 3 を削除する。

20

【 0 0 2 1 】

情報処理装置 1 によれば、特定手段 1 b により、記憶手段 1 a が参照されて、データ項目 2 a , 2 b , 2 c それぞれのうち、最新の更新データによる設定が初期設定と同一でないデータ項目 2 a , 2 c につき最新の更新データ 3 d , 3 e が特定される。処理手段 1 c により、特定手段 1 b が特定した更新データ 3 d , 3 e を用いて、データ項目 2 a , 2 b , 2 c に対する 1 つの第 2 の更新データ 4 が作成されて記憶手段 1 a に格納され、記憶手段 1 a から第 1 の更新データ群 3 が削除される。

30

【 0 0 2 2 】

これにより、更新ログのデータ量を効率的に低減できる。具体的には、データ項目 2 a , 2 b , 2 c それぞれの最新の更新データを反映した第 2 の更新データ 4 を生成して、それ以外の更新データを削除するようにしたので、更新ログのデータ量を低減できる。ここで、データ項目 2 a , 2 b , 2 c それぞれの最新の更新データのみから第 2 の更新データ 4 を生成できるのは、最新の更新データ以外の更新データが、最終的に新しい更新データによって別の設定に更新されるものだからである。このため、最新の更新データのみを参照することで、当該データ項目を同期させるために必要な情報を得ることができる。

【 0 0 2 3 】

特に、レコード 2 に対して、1 つの新たな更新データ (第 2 の更新データ 4) を生成するので、レコード 2 に対して複数存在していた更新データのデータ量を効率的に低減できる。

40

【 0 0 2 4 】

更に、最新の更新データによる設定が初期設定と異なるデータ項目 2 a , 2 c について第 2 の更新データ 4 を生成することで、第 2 の更新データ 4 の中に初期設定と同一設定のデータ項目 2 b につき無駄な更新コマンドを含めずに済む。

【 0 0 2 5 】

このため、第 2 の更新データ 4 を用いて冗長化したデータベースの同期を行えば、同期処理に掛かる時間を短縮できる。また、更新ログの最大データ量が所定量に制限されていても、第 2 の更新データ 4 の生成および第 1 の更新データ群 3 の削除を定期的に行うこと

50

で、更新ログの増加量を緩和できる。

【 0 0 2 6 】

以下に説明する第 2 の実施の形態では、情報処理装置 1 を金融窓口システムに適用する場合を例示して説明する。

[第 2 の実施の形態]

図 2 は、第 2 の実施の形態の金融窓口システムを示す図である。この金融窓口システムは、金融機関のセンタ 1 0 および営業店 2 0 における為替業務を支援する。ここで、以下では振込業務を例示して説明する。ただし、他の業務に関しても同様である。センタ 1 0 および営業店 2 0 には、金融窓口システムを構成するサーバ装置（以下、単にサーバということもある）や端末装置などの情報処理装置が設置されている。

10

【 0 0 2 7 】

この金融窓口システムは、D B (DataBase) サーバ 1 0 0 , 1 0 0 a、A P (Application) サーバ 2 0 0、W e bサーバ 3 0 0、手続端末装置 4 0 0、窓口端末装置 5 0 0 および承認端末装置 6 0 0 を含む。

【 0 0 2 8 】

D Bサーバ 1 0 0 , 1 0 0 a、A Pサーバ 2 0 0 および W e bサーバ 3 0 0 はセンタ 1 0 内のネットワーク 1 1 に接続されている。手続端末装置 4 0 0 はセンタ 1 0 内のネットワーク 1 2 に接続されている。また、ネットワーク 1 2 はネットワーク 1 1 に接続されている。更に、窓口端末装置 5 0 0 および承認端末装置 6 0 0 は営業店 2 0 内のネットワーク 2 1 に接続されている。そして、ネットワーク 1 1 , 2 1 は、ネットワーク 3 0 を介して接続されている。ネットワーク 3 0 は、インターネット、イントラネットまたは当該金融窓口システムのための専用の通信網である。

20

【 0 0 2 9 】

D Bサーバ 1 0 0 , 1 0 0 a は、金融窓口システムの処理に利用するデータを格納するためのサーバ装置である。D Bサーバ 1 0 0 , 1 0 0 a は、冗長化（ミラーリング）されている。すなわち、D Bサーバ 1 0 0 , 1 0 0 a は、一方（運用系）の更新内容を他方（待機系）へ同期する機能を有する。特に、D Bサーバ 1 0 0 , 1 0 0 a は運用系が障害により運用継続できなくなると、待機系へ業務処理を引き継ぐ。例えば、D Bサーバ 1 0 0 a が運用系であった場合に、D Bサーバ 1 0 0 a で障害が発生し、運用継続できなくなると、待機系である D Bサーバ 1 0 0 に業務処理を引き継ぐ。その際、D Bサーバ 1 0 0 は、D Bサーバ 1 0 0 での更新内容を記録した更新ログを生成する。更新ログは、D Bサーバ 1 0 0 a が復旧したときに、D Bサーバ 1 0 0 での更新内容を D Bサーバ 1 0 0 a に同期させるために用いる。

30

【 0 0 3 0 】

A Pサーバ 2 0 0 は、金融窓口システムの業務処理を実現するためのアプリケーションを実行するサーバ装置である。A Pサーバ 2 0 0 は、W e bサーバ 3 0 0 から受け付けた要求に応じて D Bサーバ 1 0 0 , 1 0 0 a を参照し、業務処理を実行する。

【 0 0 3 1 】

W e bサーバ 3 0 0 は、手続端末装置 4 0 0、窓口端末装置 5 0 0 および承認端末装置 6 0 0 から業務処理要求を受け付けるための G U I (Graphical User Interface) を提供するサーバ装置である。手続端末装置 4 0 0、窓口端末装置 5 0 0 および承認端末装置 6 0 0 の各オペレータは、各装置に設けられたブラウザによって当該 G U I を利用できる。

40

【 0 0 3 2 】

手続端末装置 4 0 0 は、振込手続を行うための情報処理装置である。手続端末装置 4 0 0 は、センタ 1 0 に駐在する手続オペレータが操作する。

窓口端末装置 5 0 0 は、顧客が記入した振込伝票を受け付ける情報処理装置である。窓口端末装置 5 0 0 は、営業店 2 0 に駐在する窓口オペレータが操作する。

【 0 0 3 3 】

承認端末装置 6 0 0 は、振込手続の継続を承認するための情報処理装置である。承認端末装置 6 0 0 は、営業店 2 0 に駐在する承認者が操作する。

50

図3は、各装置の関係を示す図である。窓口端末装置500、承認端末装置600および手続端末装置400の各オペレータは、Webサーバ300にアクセスして業務処理要求を入力できる。窓口端末装置500、承認端末装置600および手続端末装置400はWebサーバ300のGUIをブラウザにより取得して表示する。

【0034】

窓口端末装置500が表示するGUIは、振込伝票の受付完了の入力を受け付けるためのものである。

承認端末装置600が表示するGUIは、入力された振込伝票の情報に基づいて、振込手続を行ってもよい旨の承認を行うためのGUIである。

【0035】

手続端末装置400が表示するGUIは、振込手続の確定の入力を行うためのGUIである。

Webサーバ300は、窓口端末装置500、承認端末装置600および手続端末装置400から取得した各業務処理要求に応じてAPサーバ200に対するリクエスト(APリクエストと称する)を生成し、APサーバ200に送信する。Webサーバ300は、APリクエストに応じた処理応答(APレスポンスと称する)をAPサーバ200から受信すると、当該APレスポンスに基づき業務処理応答を生成し、窓口端末装置500、承認端末装置600および手続端末装置400に送信する。

【0036】

APサーバ200は、Webサーバ300から受信したAPリクエストに応じてDBサーバ100, 100aに対するリクエスト(DBリクエストと称する)を生成し、DBサーバ100, 100aに送信する。ここで、APサーバ200は、現在運用系のDBサーバに対してDBリクエストを送信する。例えば、DBサーバ100aが運用系であれば、DBサーバ100aにDBリクエストを送信する。すると、APサーバ200は、DBリクエストに応じた処理応答(DBレスポンスと称する)をDBサーバ100aから受信し、当該DBレスポンスに基づきAPレスポンスを生成してWebサーバ300に送信する。DBリクエストにはSQL文に対応する情報が含まれる。

【0037】

DBサーバ100, 100aは、APサーバ200から受信したDBリクエストに応じてデータ処理を行う。DBサーバ100, 100aは、当該データ処理の結果に基づきAPサーバ200に対する応答を生成し、APサーバ200に送信する。ここで、DBサーバ100, 100aのうち、運用系の一方が当該処理を行う。待機系の他方には、運用系で実行したデータ処理の結果が同期される。

【0038】

以下の説明では、DBサーバ100aは障害により停止している場合を想定する。すなわち、金融窓口システムがDBサーバ100のみで縮退運用されている場合を想定する。ただし、DBサーバ100が障害で停止し、DBサーバ100aのみで縮退運用されている場合も以下の説明と同様である。

【0039】

図4は、振込ワークフローを例示する図である。振込ワークフロー50は、金融窓口システムの振込処理の順序を示す。振込ワークフロー50には、各拠点(営業店20およびセンタ10)における、業務処理要求の発生元(要求元)の装置、当該要求に伴う処理および当該処理後の伝票状態が示されている。ここで、伝票状態は、DBサーバ100, 100aで管理される振込伝票の状態を示す。振込ワークフロー50による処理順序は、次の通りである。

【0040】

まず、顧客が金融機関に対して振込依頼を行うための振込伝票を作成する。振込ワークフロー50では、要求元が“(顧客) ”、処理が“(振込依頼) ”、伝票状態が“(作成) ”と括弧で括られている。これは、当該手順が顧客によって行われるものであり、金融窓口システムが行う手順ではないためである。ただし、金融窓口システムによる処理の起

10

20

30

40

50

点となる入力データは当該伝票に基づいて生成されるため、それが分かり易いように図示している。以下、それ以降の手順を説明する。なお、DBサーバ100の処理は、上述したようにWebサーバ300およびAPサーバ200を介したDBリクエストに基づき実行される。

【0041】

(1) 窓口端末装置500は、振込伝票の情報の登録要求をWebサーバ300に送信する。すると、DBサーバ100は当該伝票の情報を登録する。DBサーバ100は、伝票状態を“受付”に設定する。

【0042】

(2) 窓口端末装置500は、振込伝票による振込の承認依頼要求をWebサーバ300に送信する。すると、DBサーバ100は、伝票状態を“承認待ち”に設定する。これによって、承認端末装置600による承認が可能となる。

10

【0043】

(3) 承認端末装置600は、振込伝票の承認の設定要求をWebサーバ300に送信する。すると、DBサーバ100は伝票状態を“承認”に設定する。

(4) 承認端末装置600は、振込伝票につき振込手続の委譲要求をWebサーバ300に送信する。すると、DBサーバ100は、伝票状態を“委譲”に設定する。

【0044】

(5) 手続端末装置400は、振込伝票による振込手続の実行要求をWebサーバ300に送信する。すると、DBサーバ100は、伝票状態を“手続中”に設定する。そして、APサーバ200により、他の支店や他の金融機関等との間で振込処理が実行される。

20

【0045】

(6) APサーバ200は、振込処理が完了すると、その旨をDBサーバ100に通知する。すると、DBサーバ100は、伝票状態を“完了”に設定する。

以上の順序で振込処理が行われ、DBサーバ100上の電子伝票の状態が更新される。DBサーバ100は、このように順次変更される電子伝票の更新の履歴を更新ログとして記録する。DBサーバ100aが復旧した際に、当該更新ログを用いてデータの同期を図るためである。その際、DBサーバ100は、当該更新ログのデータ量を効率的に低減する。以下では、そのようなDBサーバ100の構成を詳細に説明する。まず、DBサーバ100が備えるデバイス(ハードウェア)の構成を説明する。

30

【0046】

図5は、DBサーバのハードウェア構成を示す図である。DBサーバ100は、CPU(Central Processing Unit)101、ROM(Read Only Memory)102、RAM(Random Access Memory)103、HDD(Hard Disk Drive)104、グラフィック処理装置105、入力インタフェース106、記録媒体読取装置107および通信インタフェース108を有する。

【0047】

CPU101は、OS(Operating System)プログラムやアプリケーションプログラムを実行して、DBサーバ100全体を制御する。

ROM102は、DBサーバ100の起動時に実行されるBIOS(Basic Input / Output System)プログラムなどの所定のプログラムを記憶する。ROM102は、書き換え可能な不揮発性メモリであってもよい。

40

【0048】

RAM103は、CPU101が実行するOSプログラムやアプリケーションプログラムの少なくとも一部を一時的に記憶する。また、RAM103は、CPU101の処理に用いられるデータの少なくとも一部を一時的に記憶する。

【0049】

HDD104は、OSプログラムやアプリケーションプログラムを記憶する。また、HDD104は、CPU101の処理に用いられるデータを記憶する。なお、HDD104に代えて(または、HDD104と併せて)、SSD(Solid State Drive)など他の種

50

類の不揮発性の記憶装置を用いてもよい。

【0050】

グラフィック処理装置105は、モニタ13に接続される。グラフィック処理装置105は、CPU101からの命令に従って、画像をモニタ13に表示させる。

入力インタフェース106は、キーボード14やマウス15などの入力デバイスに接続される。入力インタフェース106は、入力デバイスから送られる入力信号をCPU101に出力する。

【0051】

記録媒体読取装置107は、記録媒体16に格納されたデータを読み取る読取装置である。記録媒体16には、例えば、DBサーバ100に実行させるプログラムが記録されている。DBサーバ100は、例えば、記録媒体16に記録された情報処理プログラムを実行することで、後述するような更新ログの圧縮機能を実現することができる。すなわち、情報処理プログラムはコンピュータ読み取り可能な記録媒体16に記録して配布することが可能である。

10

【0052】

記録媒体16としては、例えば、磁気記録装置、光ディスク、光磁気記録媒体、半導体メモリを使用できる。磁気記録装置には、HDD、フレキシブルディスク(FD)、磁気テープなどがある。光ディスクには、CD(Compact Disc)、CD-R(Recordable)/RW(ReWritable)、DVD(Digital Versatile Disc)、DVD-R/RW/RAMなどがある。光磁気記録媒体には、MO(Magneto-Optical disk)などがある。半導体メモリには、USB(Universal Serial Bus)メモリなどのフラッシュメモリがある。

20

【0053】

通信インタフェース108は、ネットワーク11に接続される。通信インタフェース108は、ネットワーク11を介してDBサーバ100a、APサーバ200、Webサーバ300、手続端末装置400、窓口端末装置500および承認端末装置600とデータ通信を行える。

【0054】

なお、ネットワーク11やネットワーク30等に接続された他のサーバ装置(図示せず)に情報処理プログラムを格納しておいてもよい。その場合、DBサーバ100は、当該サーバ装置から情報処理プログラムをダウンロードして実行することもできる。

30

【0055】

また、DBサーバ100a、APサーバ200、Webサーバ300、手続端末装置400、窓口端末装置500および承認端末装置600もDBサーバ100と同様のハードウェア構成により実現できる。

【0056】

図6は、DBサーバの機能構成を示す図である。DBサーバ100は、伝票データベース110、トランザクションログ記憶部120、更新ログ記憶部130、最新レコード記憶部140、通信部150、データベース管理部160、解析処理部170、圧縮処理部180および同期処理部190を有する。これらのユニットの機能は、CPU101が情報処理プログラムを実行することによりDBサーバ100上に実現される。ただし、これらのユニットの機能の全部または一部を専用のハードウェアで実現してもよい。

40

【0057】

伝票データベース110は、電子伝票を保存し、管理するための記憶領域である。伝票データベース110は、1つの電子伝票を、複数の設定項目を含む1つのレコードとして管理する。

【0058】

トランザクションログ記憶部120は、トランザクションログを記憶する。トランザクションログは、伝票データベース110に対する全てのトランザクションを記録したものである。トランザクションログは、伝票データベース110に対する個別のトランザクションのロールバックやロールフォワードを行うために用いられる。

50

【0059】

更新ログ記憶部130は、更新ログを記憶する。更新ログは、DBサーバ100aが障害により停止状態にあるときにデータベース管理部160により取得される。更新ログは、DBサーバ100aが障害から復旧した際に、DBサーバ100, 100aで同期をとるために用いられる。

【0060】

最新レコード記憶部140は、最新レコード管理データを記憶する。最新レコード管理データは、伝票データベース110に記録されたレコードに含まれる各設定項目につき、最新の状態を特定するために用いる情報である。

【0061】

通信部150は、APサーバ200と通信する。通信部150は、APサーバ200から受信したDBリクエストをデータベース管理部160に出力する。通信部150は、データベース管理部160から取得したDBレスポンスをAPサーバ200に送信する。

【0062】

データベース管理部160は、通信部150から取得したDBリクエストを解析してSQL文を取得(あるいは生成)する。データベース管理部160は、当該SQL文を実行して伝票データベース110を操作する。操作内容は、例えば、更新(UPDATE)、削除(DELETE)および挿入(INSERT)である。

【0063】

解析処理部170は、更新ログ記憶部130に記憶された更新ログを参照して、最新レコード管理データを生成し、最新レコード記憶部140に格納する。

圧縮処理部180は、最新レコード記憶部140に記憶された最新レコード管理データを参照して更新ログの圧縮処理を行う。

【0064】

同期処理部190は、DBサーバ100aが障害から復旧した際に、更新ログ記憶部130に記憶された更新ログを参照して、DBサーバ100aが備える伝票データベースを伝票データベース110に同期させる。

【0065】

なお、DBサーバ100aもDBサーバ100と同様の機能構成である。

図7は、伝票管理テーブルのデータ構造例を示す図である。伝票管理テーブル111は、伝票データベース110に格納され、データベース管理部160によって更新される。伝票管理テーブル111は電子伝票を管理するためのデータである。伝票管理テーブル111には、レコード番号、エントリ店舗番号、伝票通番、伝票状態、取出中フラグおよびメディアファイルを示す項目が設けられている。各項目の横方向に並べられた情報同士が互いに関連付けられて、1つの電子伝票の情報を示す。

【0066】

レコード番号の項目には、レコードを識別するための番号が設定される。エントリ店舗番号の項目には、営業店20を識別するための識別情報が設定される。伝票通番の項目には、電子伝票に付された通し番号が設定される。伝票状態の項目には、振込ワークフロー50上の伝票状態が設定される。取出中フラグの項目には、当該フラグがONまたはOFFの何れであるかを示す情報が設定される。ここで、取出中フラグとは、振込ワークフロー50における何れかの処理により利用を行っているため、他の処理によって更新できない状態(排他状態)であることを示すフラグである。このようなフラグは、排他フラグと呼ばれることもある。メディアファイルの項目には、振込伝票の記入内容を画像化したファイルが設定される(伝票管理テーブル111ではファイル名で示している)。

【0067】

例えば、伝票管理テーブル111には、レコード番号が“1”、エントリ店舗番号が“E001”、伝票通番が“D0001”、伝票状態が“受付”、取出中フラグが“OFF”、メディアファイルが“D0001.dat”という情報が設定されている。

【0068】

10

20

30

40

50

これは、当該電子伝票を示すレコード番号が“ 1 ”であること、営業店 20 で登録された伝票であること、伝票通番が“ D 0 0 0 1 ”であること、伝票状態が振込ワークフロー 50 における“ 受付 ”の状態であることを示す。また、取出中フラグが“ O F F ”であるので、何れの処理にも用いられていないことを示す。更に、当該電子伝票に対応するメディアファイルが“ D 0 0 0 1 . d a t ”として、伝票管理テーブル 1 1 1 に格納されていることを示す。

【 0 0 6 9 】

また、例えば、伝票管理テーブル 1 1 1 には、レコード番号が“ 2 ”、エントリ店舗番号が“ E 0 0 1 ”、伝票通番が“ D 0 0 0 2 ”、伝票状態が“ 承認待ち ”、取出中フラグが“ O N ”、メディアファイルが“ D 0 0 0 2 . d a t ”という情報が設定される。

10

【 0 0 7 0 】

これは、当該電子伝票を示すレコード番号が“ 2 ”であること、営業店 20 で登録された伝票であること、伝票通番が“ D 0 0 0 2 ”であること、伝票状態が振込ワークフロー 50 における“ 承認待ち ”の状態であることを示す。また、取出中フラグが“ O N ”であるので、何れかの処理（ここでは、振込ワークフロー 50 における（ 2 ）または（ 3 ）の何れか）に利用されており、当該処理以外の処理には、更新が禁止された状態であることを示す。更に、当該電子伝票に対応するメディアファイルが“ D 0 0 0 2 . d a t ”として、伝票管理テーブル 1 1 1 に格納されていることを示す。

【 0 0 7 1 】

ここで、以下の説明では電子伝票を示すそれぞれのレコードを伝票管理レコードと称することとする。

20

図 8 は、更新ログファイルのデータ構造例を示す図である。更新ログファイル 1 3 1 は、データベース管理部 1 6 0 により生成され、更新ログ記憶部 1 3 0 に格納される。更新ログファイル 1 3 1 には、複数の更新データが記述される。更新ログファイル 1 3 1 では、更新データに対応付けて、当該更新データを識別するための項番を便宜的に表記している。項番は、時系列に昇順に付されており、番号が小さい程古く、番号が大きいほど新しい。なお、各更新データにつき、テーブル名は“ 伝票管理 ”の表記によって伝票管理テーブル 1 1 1 を指定しているとする。

【 0 0 7 2 】

更新データには、データベース管理部 1 6 0 による伝票データベース 1 1 0 の更新内容が記述される。更新ログファイル 1 3 1 では、1 つの電子伝票に関する新たなレコードの挿入（ I N S E R T ）から始まって、振込ワークフロー 50 の（ 1 ）～（ 6 ）（以下、単に“（ 1 ）の処理”等と表記する）に示した一連の処理に対応する更新データを例示している。

30

【 0 0 7 3 】

具体的には、新たな伝票の登録（レコードの挿入）は項番 1 の更新データに対応する。（ 1 ）の処理は項番 2 ～ 4 の更新データに対応する。（ 2 ）の処理は項番 5 ～ 7 の更新データに対応する。（ 3 ）の処理は項番 8 ～ 1 0 の更新データに対応する。（ 4 ）の処理は項番 1 1 ～ 1 3 の更新データに対応する。（ 5 ）の処理は項番 1 4 ～ 1 6 の更新データに対応する。（ 6 ）の処理は項番 1 7 ～ 1 9 の更新データに対応する。

40

【 0 0 7 4 】

ここで、（ 1 ）の処理に対応する項番 2 ～ 4 の更新データが示す内容について説明する。更新対象のレコードのレコード番号は“ 1 ”とする。（ 1 ）の処理では、まず、項番 2 の更新データによって、取出中フラグが“ O N ”に設定される。これによって、当該レコードが排他状態となる。次に、項番 3 の更新データによって、伝票状態が“ 受付 ”に設定される。更に、項番 4 の更新データによって、取出中フラグが“ O F F ”に設定される。これによって、当該レコードの排他状態が解除される。

【 0 0 7 5 】

（ 2 ）～（ 6 ）の処理に関しても（ 1 ）の処理と同様にして、伝票状態が更新される。すなわち、まず、当該レコードを排他状態にし、伝票状態を更新して、排他状態を解除す

50

る。

【0076】

また、更新ログファイル131において、項番1の更新データは更新種別が“INSERT”（新規追加）であるから更新ログファイル131の取得を開始した時点におけるレコード番号“1”の伝票管理レコードの状態を示している。すなわち、当該項番1の更新データは当該レコードに含まれる各データ項目の初期設定である。

【0077】

一方、項番1の更新データの更新種別が“UPDATE”である場合もある。その場合、項番1の更新データは“UPDATE”対象のデータ項目の初期設定を示すことになる。そして、他のデータ項目の初期設定は、当該他のデータ項目が最初に更新対象となったときに取得され記憶される。よって、この場合、初期設定を示す更新データは複数存在することになる。

10

【0078】

図9は、最新レコードテーブルのデータ構造例を示す第1の図である。最新レコードテーブル141は、解析処理部170によって生成され、最新レコード記憶部140に格納される。最新レコードテーブル141a, 141bは、解析処理部170による最新レコードテーブル141の更新の変遷を示すものである。最新レコードテーブルには、テーブル名、レコード番号、削除フラグおよび更新内容の項目が設けられている。各項目の横方向に並べられた情報同士が互いに関連付けられて、1つのレコードの最新の状態を示す。

【0079】

テーブル名の項目には、テーブルの名称が設定される。レコード番号の項目には、テーブル内のレコードを識別するための番号が設定される。削除フラグの項目には、当該レコードが削除（DELETE）されたものが否かを示す削除フラグが設定される。削除フラグは、当該レコードが削除されたものであれば“ON”、それ以外では“OFF”が設定される。更新内容の項目には、データの項目名と当該データ項目の更新値が設定される。更新内容の項目は、データ項目名ごとに設けられる。このため、複数のデータ項目の更新値を管理する場合には、更新内容の項目も複数設けられる。最新レコードテーブル141a, 141bでは、更新内容（A）,（B）の項目を設けて、複数の更新内容を区別する。

20

【0080】

以下、最新レコードテーブル141の更新の変遷について具体的に説明する。

最新レコードテーブル141は、更新ログファイル131の項番2の更新データに対応する。すなわち、最新レコードテーブル141は、伝票管理テーブル111のレコード番号“1”のレコードにつき、項目名“取出中フラグ”を更新値“ON”に設定した旨を記録したものである。

30

【0081】

最新レコードテーブル141aは、更新ログファイル131の項番3の更新データに対応する。すなわち、最新レコードテーブル141aは、伝票管理テーブル111のレコード番号“1”のレコードにつき、項目名“伝票状態”を更新値“受付”に設定した旨を記録したものである。

40

【0082】

最新レコードテーブル141bは、更新ログファイル131の項番4の更新データに対応する。すなわち、最新レコードテーブル141bは、伝票管理テーブル111のレコード番号“1”のレコードにつき、項目名“取出中フラグ”を更新値“OFF”に設定した旨を記録したものである。

【0083】

以下、更新ログファイル131の項番5～19についても、同様にして最新レコードテーブル141bを更新していく。

図10は、最新レコードテーブルのデータ構造例を示す第2の図である。最新レコードテーブル141nは、更新ログファイル131の項番19の更新データに対応する。すな

50

わち、最新レコードテーブル 1 4 1 n は、伝票管理テーブル 1 1 1 のレコード番号 “ 1 ” のレコードの最新の状態を特定したものである。

【 0 0 8 4 】

次に、以上の構成の DB サーバ 1 0 0 の処理手順を説明する。以下に示す処理は、例えば縮退運用時に定期的に行われる。

図 1 1 は、更新ログ圧縮処理を示すフローチャートである。以下、図 1 1 に示す処理をステップ番号に沿って説明する。

【 0 0 8 5 】

[ステップ S 1] 解析処理部 1 7 0 は、更新ログ記憶部 1 3 0 に更新ログファイル 1 3 1 が有るか否かを判定する。更新ログファイル 1 3 1 が有る場合、処理をステップ S 2 に進める。更新ログファイル 1 3 1 が無い場合、処理を終了する。

10

【 0 0 8 6 】

[ステップ S 2] 解析処理部 1 7 0 は、更新ログファイル 1 3 1 から未処理の更新データのうち最も古い更新データを 1 つ取得する。

[ステップ S 3] 解析処理部 1 7 0 は、更新データに記述された更新種別を判定する。“ U P D A T E ” の場合、処理をステップ S 4 に進める。“ D E L E T E ” の場合、処理をステップ S 5 に進める。“ I N S E R T ” の場合、処理をステップ S 6 に進める。

【 0 0 8 7 】

[ステップ S 4] 解析処理部 1 7 0 は、最新レコード記憶部 1 4 0 に記憶された最新レコードテーブル 1 4 1 に更新内容を上書き記録する。

20

[ステップ S 5] 解析処理部 1 7 0 は、最新レコード記憶部 1 4 0 に記憶された最新レコードテーブル 1 4 1 の削除フラグの項目を “ O N ” に設定する。

【 0 0 8 8 】

[ステップ S 6] 解析処理部 1 7 0 は、更新ログファイル 1 3 1 に含まれる全ての更新データを処理済みであるか否かを判定する。全ての更新データを処理済みである場合、処理をステップ S 7 に進める。なお、処理をステップ S 7 に進める時点で最新レコード記憶部 1 4 0 には最新レコードテーブル 1 4 1 n が格納されているとする。未処理の更新データが有る場合、処理をステップ S 2 に進める。

【 0 0 8 9 】

[ステップ S 7] 圧縮処理部 1 8 0 は、最新レコードテーブル 1 4 1 n を参照して、レコードごとに単一の更新データを生成する。単一の更新データとは、当該レコードに対して 1 つの処理コマンド (“ U P D A T E ” や “ D E L E T E ”) で、各データ項目の更新を行う更新データである。このとき、圧縮処理部 1 8 0 は最新レコードテーブル 1 4 1 n の更新値が初期設定と異なるデータ項目のみを、更新データに含める。レコードの更新を行う場合 (削除フラグが “ O F F ” の場合) 、更新種別 “ U P D A T E ” の更新データを生成する。レコードの削除を行う場合 (削除フラグが “ O N ” の場合) 、更新種別 “ D E L E T E ” の更新データを生成する。圧縮処理部 1 8 0 は、生成した更新データを更新ログファイル 1 3 1 に記述する。

30

【 0 0 9 0 】

[ステップ S 8] 圧縮処理部 1 8 0 は、更新ログ記憶部 1 3 0 に記憶されたステップ S 7 で生成した更新データ以外の更新データ (初期設定の更新データを除く) を削除する。

40

このようにして、解析処理部 1 7 0 および圧縮処理部 1 8 0 は更新ログファイル 1 3 1 のデータ量を低減する。

【 0 0 9 1 】

図 1 2 は、圧縮処理後の更新ログファイルを例示する第 1 の図である。更新ログファイル 1 3 1 a は、更新ログファイル 1 3 1 および最新レコードテーブル 1 4 1 n に基づいて、圧縮処理部 1 8 0 により生成されたものであり、図 1 1 のステップ S 8 の圧縮処理部 1 8 0 による処理後の更新ログである。更新ログファイル 1 3 1 と比較すると、項番 2 0 の更新データが追加されて、項番 2 ~ 1 9 のデータが削除されていることが分かる。

【 0 0 9 2 】

50

特に、項番20の更新データでは、対象の伝票管理レコードにつき伝票状態を“完了”に更新する処理が記述されている。これは、最新レコードテーブル141nの更新内容(B)に対応する。一方、項番20の更新データでは、最新レコードテーブル141nの更新内容(A)に対応する処理は記述されていない。これは、項番1の更新データに示される取出中フラグ“OFF”と、最新レコードテーブル141nの更新内容(A)で特定された取出中フラグの最新の設定“OFF”と、が同一だからである。すなわち、更新の必要がないため、当該データ項目に関する処理の記述を省ける。

【0093】

次に、最新レコードテーブル141nの他の例と、それに応じて圧縮処理部180により生成される更新ログファイル131aの他の例とを具体的に説明する。

図13は、圧縮処理後の更新ログファイルを例示する第2の図である。まず、最新レコードテーブル141xは、最新レコードテーブル141nの他の例を示す。具体的には、更新内容(C)の項目が追加されている。当該更新内容(C)の項目には、項目名“伝票通番”および更新値“D1001”が設定されている。これは、レコード番号“1”に対して、伝票通番が“D1001”に変更されていることを示す。更新内容(C)は、圧縮前の更新ログファイルから、更新内容(B)とは別個の更新データにより特定されたものである。

【0094】

圧縮処理部180は、圧縮処理前の更新ログファイルと最新レコードテーブル141xとに基づいて更新ログファイル131bを生成する。ここで、圧縮処理前の更新ログファイルでは、初期設定(項番1)の更新データの更新種別が“INSERT”、取出中フラグが“OFF”、伝票状態が“完了”以外、伝票通番が“D1001”以外であったとする。この場合、更新種別が“UPDATE”なので、上述したように項番1の更新データも削除できる。

【0095】

更新ログファイル131bには、項番1の更新データに加えて、項番31の更新データが記述されている。当該項番31の更新データでは、対象の伝票管理レコードにつき伝票状態を“完了”に、伝票通番を“D1001”に更新する処理が記述されている。これは、最新レコードテーブル141xの更新内容(B)、(C)に対応する。また、項番31の更新データでは最新レコードテーブル141xの更新内容(A)に対応する処理は記述されていない。これは、初期設定の取出中フラグ“OFF”と、最新レコードテーブル141xの更新内容(A)で特定された取出中フラグの最新の設定“OFF”と、が同一だからである。

【0096】

このように、圧縮処理部180は圧縮処理前には複数の更新データで記述されていた複数の更新内容を単一の更新データにまとめて、更新ログファイルに記述する。そして、当該伝票管理レコードの更新内容を記述していた複数の更新データを圧縮前の更新ログファイルから削除する。その結果、更新ログファイル131bが生成される。

【0097】

これにより、更新ログファイルのデータ量を効率的に低減できる。特に、複数の更新内容を単一の更新データにまとめる。このため、当該更新ログファイル131bを用いて同期処理を行えば、圧縮前の更新ログファイルを用いて同期を行う場合に比べて、更新処理の回数を低減できる。よって、同期処理を高速に行える。

【0098】

このとき、初期設定と最新の設定とで相違がないデータ項目に関しては、更新データに更新処理を記述しない。よって、更新ログファイルのデータ量を一層低減できる。また、同期処理において当該データ項目に対する更新処理を行わずに済む。よって、同期処理を一層高速に行える。

【0099】

このような制御は、更新ログファイル中に、取出中フラグなどの排他フラグのように、

10

20

30

40

50

データ処理に用いる時だけ更新し、データ処理が完了すると元の設定に戻すようなデータ項目が含まれる場合に特に有効である。なぜなら、このような排他フラグは、一連のデータ処理において頻繁に更新されるにも関わらず、最終的には当初の設定値に戻される。したがって、これら全ての更新処理を同期処理の際に再現すると、処理に無駄が生じ易いためである。

【0100】

図14は、圧縮処理後の更新ログファイルを例示する第3の図である。まず、最新レコードテーブル141yは、最新レコードテーブル141nの他の例を示す。具体的には、レコード番号“2”の伝票管理レコードに対して削除フラグが“ON”に設定されている。これは、レコード番号“2”に対して、“DELETE”の処理が行われたことを示す。なお、圧縮処理前の更新ログファイルには、レコード番号“2”の伝票管理レコードにつき、“UPDATE”の処理が複数記述されていたものとする。初期設定も“UPDATE”の更新種別を含む複数の更新データにより取得されていたものとする。

10

【0101】

圧縮処理部180は、圧縮処理前の更新ログファイルと最新レコードテーブル141yとに基づいて、更新ログファイル131cを生成する。更新ログファイル131cには、項番41の更新データが記述されている。当該項番41の更新データでは、対象の伝票管理レコードを削除する処理が記述されている。

【0102】

このように、圧縮処理部180は圧縮処理前には複数の更新データで複数の更新内容が記述されていたとしても、削除処理が行われた場合には当該削除処理を記述した更新データのみを生成し、他の更新データを削除する。削除処理を記述した更新データ以外の他の更新データを削除できるのは、削除された伝票管理レコードに対しては、その後新たな処理に用いられることは無いと考えられるからである。

20

【0103】

これにより、更新ログファイルのデータ量を一層効率的に低減できる。具体的には、削除処理以外の処理を示す更新データを削除するので、当該削除した更新データ分のデータ量を低減できる。また、更新ログファイル131cを用いれば、同期処理において、当該伝票管理レコードに対する削除処理以外の処理を行わずに済む。よって、同期処理を一層高速に行える。

30

【0104】

特に、レコードの削除処理用の更新データを生成する際には、更新ログファイル131cに示したように、更新種別“UPDATE”で記述された更新データを初期設定も含めて削除してもよい（更新種別“DELETE”で記述された更新データが残る）。削除されるレコードにつき同期時に更新処理を再現するのは無駄だからである。

【0105】

また、以上に示した圧縮処理を定期的に行えば、更新ログファイルの増加量を緩和できる。よって、更新ログファイルの最大データ量が所定量に制限されていたとしても、当該ファイルのデータ量が所定量に達するまでに時間的な余裕が得られる。その結果、縮退運転の稼働時間や保守用の時間等を確保し易くなる。

40

【0106】

以上、本発明の情報処理プログラム、情報処理装置および情報処理方法を図示の実施の形態に基づいて説明したが、これらに限定されるものではなく、各部の構成は同様の機能を有する任意の構成のものに置換することができる。また、他の任意の構成物や工程が付加されてもよい。更に、前述した実施の形態のうちの任意の2以上の構成（特徴）を組み合わせたものであってもよい。

【符号の説明】

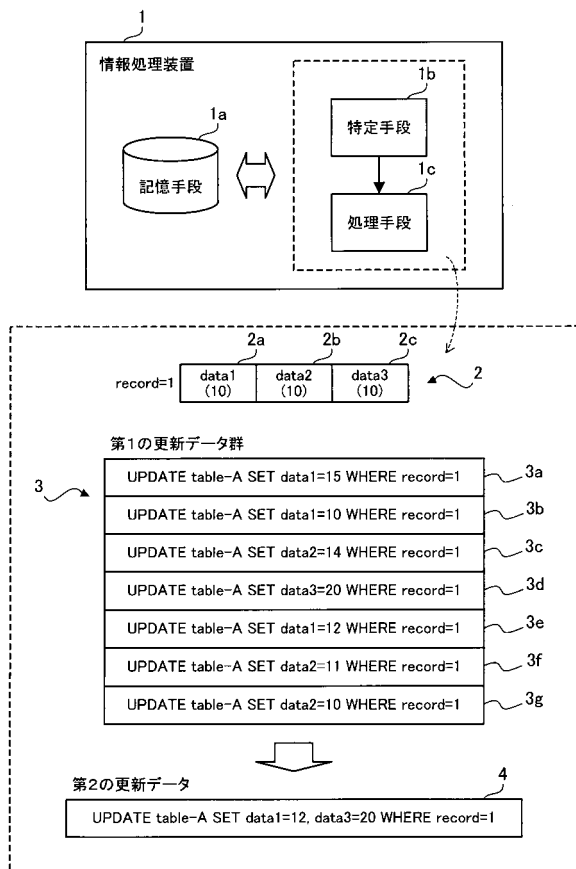
【0107】

- 1 情報処理装置
- 1a 記憶手段

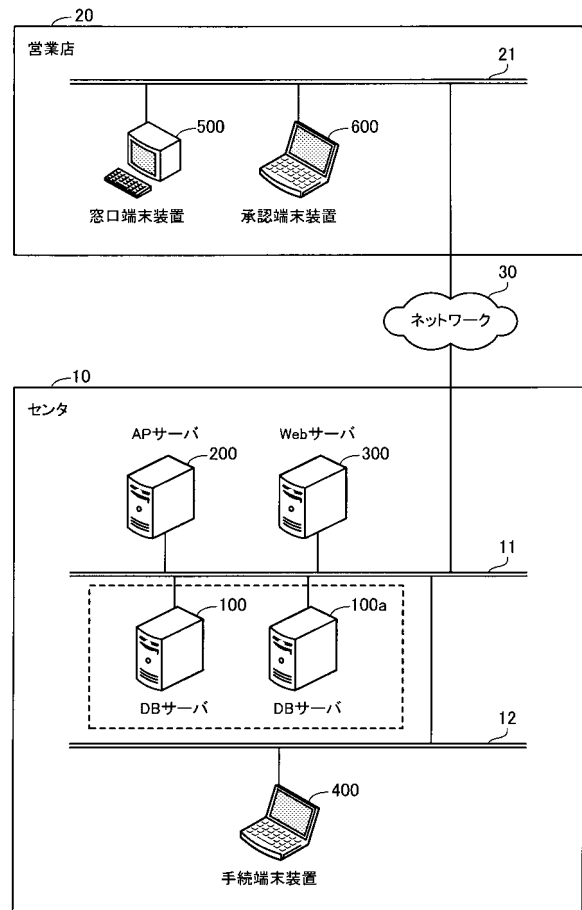
50

- 1 b 特定手段
- 1 c 処理手段
- 2 レコード
- 2 a , 2 b , 2 c データ項目
- 3 第1の更新データ群
- 3 a , 3 b , 3 c , 3 d , 3 e , 3 f , 3 g 第1の更新データ
- 4 第2の更新データ

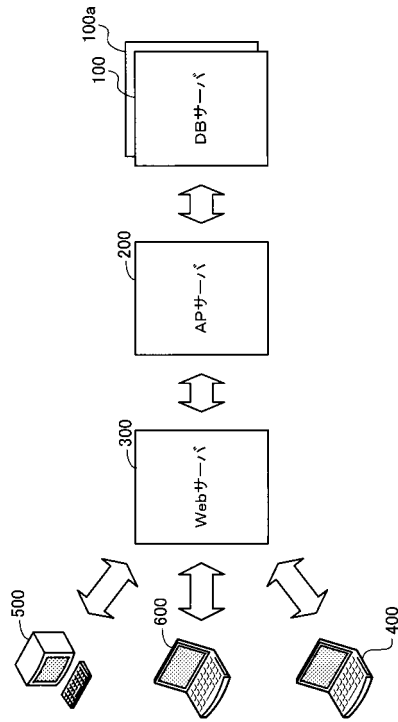
【 図 1 】



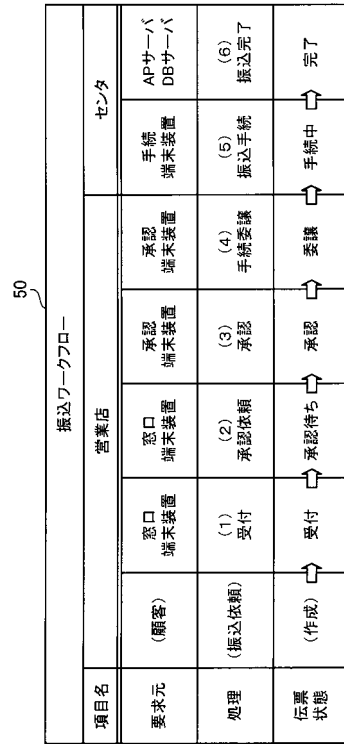
【 図 2 】



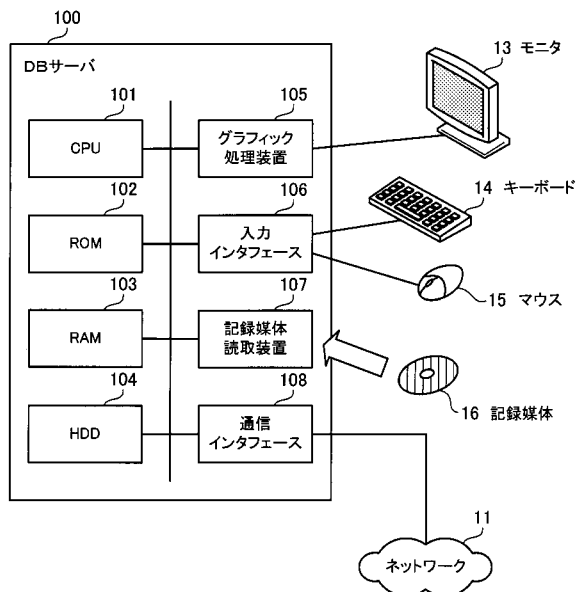
【 図 3 】



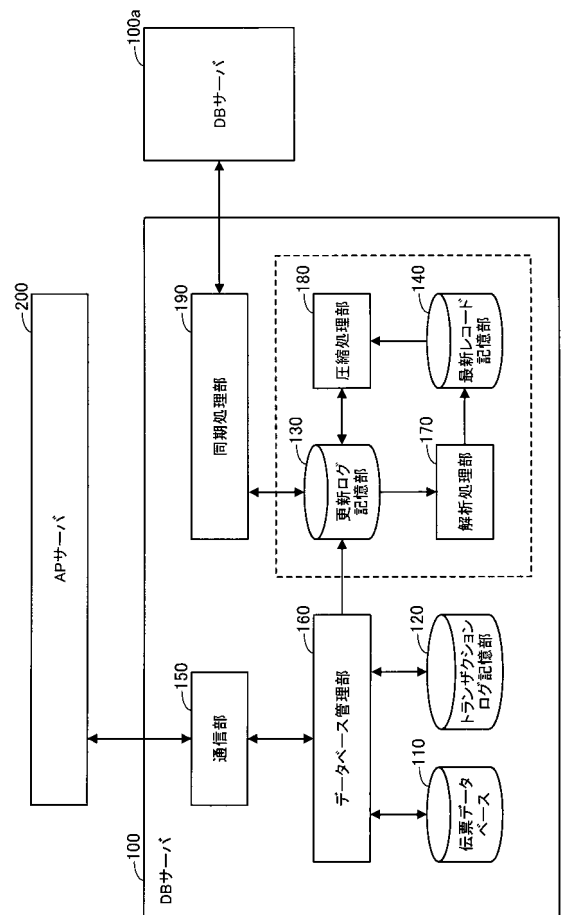
【 図 4 】



【 図 5 】



【 図 6 】



【 図 7 】

111

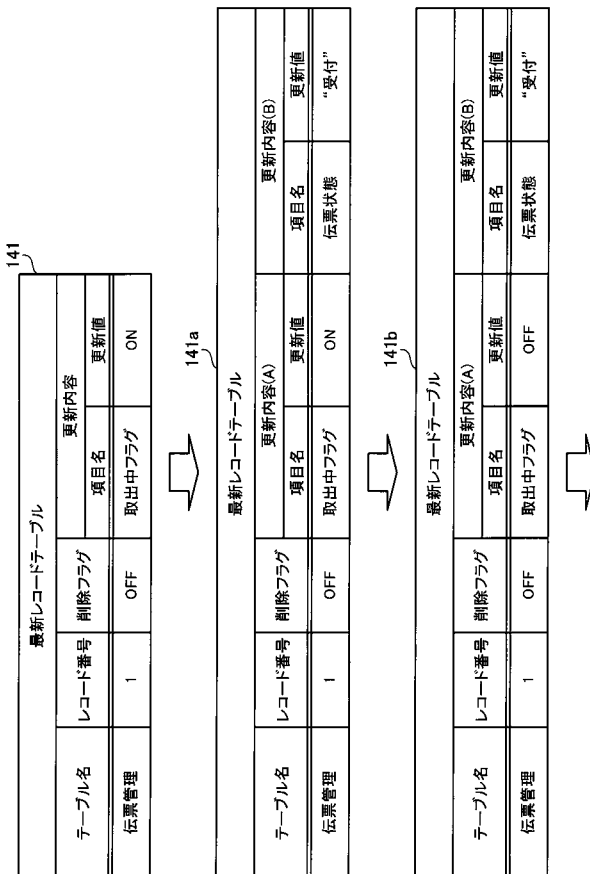
伝票管理テーブル					
レコード番号	エントリ店舗番号	伝票通番	伝票状態	取出中フラグ	メディアファイル
1	E001	D0001	受付	OFF	D0001.dat
2	E001	D0002	承認待ち	ON	D0002.dat
...

【 図 8 】

131

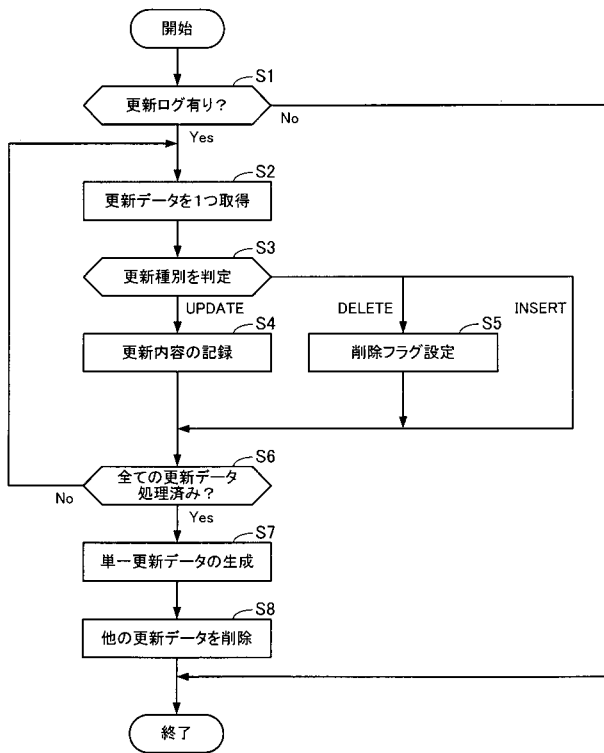
更新ログファイル	
項番	更新データ
1	INSERT INTO 伝票管理 VALUE (レコード番号=1,取出中フラグ=OFF,...)
2	UPDATE 伝票管理 SET 取出中フラグ=ON where レコード番号=1
3	UPDATE 伝票管理 SET 伝票状態="受付" where レコード番号=1
4	UPDATE 伝票管理 SET 取出中フラグ=OFF where レコード番号=1
5	UPDATE 伝票管理 SET 取出中フラグ=ON where 伝票状態="受付"
6	UPDATE 伝票管理 SET 伝票状態="承認待ち" where レコード番号=1
7	UPDATE 伝票管理 SET 取出中フラグ=OFF where レコード番号=1
8	UPDATE 伝票管理 SET 取出中フラグ=ON where 伝票状態="承認待ち"
9	UPDATE 伝票管理 SET 伝票状態="承認" where レコード番号=1
10	UPDATE 伝票管理 SET 取出中フラグ=OFF where レコード番号=1
11	UPDATE 伝票管理 SET 取出中フラグ=ON where 伝票状態="承認"
12	UPDATE 伝票管理 SET 伝票状態="委譲" where レコード番号=1
13	UPDATE 伝票管理 SET 取出中フラグ=OFF where レコード番号=1
14	UPDATE 伝票管理 SET 取出中フラグ=ON where 伝票状態="委譲"
15	UPDATE 伝票管理 SET 伝票状態="手続中" where レコード番号=1
16	UPDATE 伝票管理 SET 取出中フラグ=OFF where レコード番号=1
17	UPDATE 伝票管理 SET 取出中フラグ=ON where 伝票状態="手続中"
18	UPDATE 伝票管理 SET 伝票状態="完了" where レコード番号=1
19	UPDATE 伝票管理 SET 取出中フラグ=OFF where レコード番号=1

【 図 9 】



【 図 10 】

【 図 1 1 】

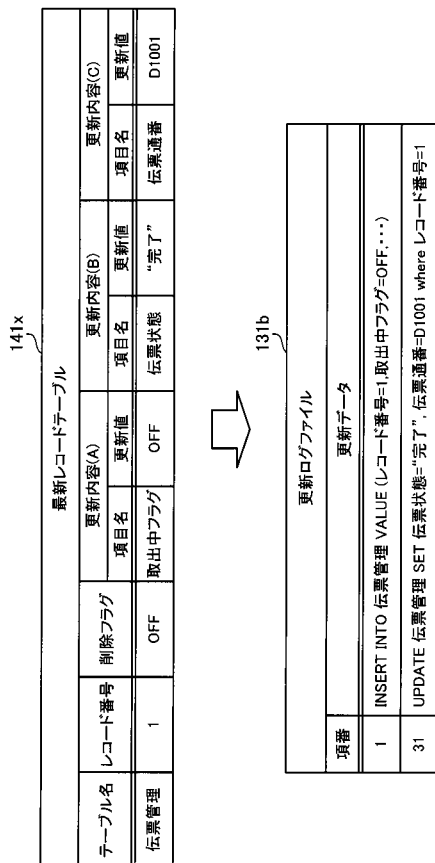


【 図 1 2 】

131a

更新ログファイル	
項番	更新データ
1	INSERT INTO 伝票管理 VALUE (レコード番号=1,取出中フラグ=OFF,...)
20	UPDATE 伝票管理 SET 伝票状態="完了" where レコード番号=1

【 図 1 3 】



【 図 1 4 】

