

CS シリーズ (OS 2200) データベースの有効活用

——DataExtractor/IX (更新差分抽出) によるレプリケーション技術

Effective Utilization of CS Series (OS 2200) Database

——Realization of Replication by DataExtractor/IX (Change Data Capture)

実 藤 英 昭 , 中 山 陽 太 郎

要 約 近年, 汎用機上の基幹システムとオープン系システム間のデータ連携を実現するためにさまざまな技術が提案されている。汎用機をデータソースとするレプリケーション機能は, データウェアハウス構築支援としての役割に留まらず, オープン系システムへのデータ供給技術として期待される。

DataExtractor/IX(更新差分抽出)は, CS シリーズ及び HMP/IX シリーズ上の DMS 1100 データベースの更新ログをオープン・プラットフォーム上へ転送し, リレーショナル・データベースに反映することによって動的レプリケーションを実現する。

本稿では, レプリケーション機能を中心に DataExtractor/IX (更新差分抽出) の機能の紹介を行う。

Abstract In these days, various techniques for data transfer between mission-critical systems on mainframes and other applications on the open platform systems have been proposed. The data replication capability that uses mainframes as data resource has not only a role for DWH(Dataware House) building support but also the generic solution for the data provision to the open platform systems. DataExtractor/IX (Changed Data Capture) collects changes made to DMS1100 database, and transfers and reflects them to the relational databases on the open platform to implement the dynamic data replication.

This report mainly explains the functionality of DataExtractor/IX(CDC) focusing on data replication.

1. はじめに

汎用機上に構築された基幹システムと Web アプリケーションとの間でのデータ連携を実現するために, 多様な技術が提案されている。汎用機ホストをデータソースとするレプリケーション技術は, 従来のデータウェアハウス構築支援としての役割に留まらず, オープン環境へのデータ供給の基盤技術として捉えられるようになった。しかし, CS シリーズ(OS 2200)のネットワーク型データベースである UDS DMS 1100^{*1} 上のデータを, オープン・プラットフォーム (Windows/UNIX) 上のリレーショナル・データベース (RDB) に展開するためには, データ形式の相違や文字コード変換, データ抽出方法といった, さまざまなレベルの問題を解決しなければならない。また開発者は OS 2200 とオープン環境の双方にわたるデータベースの知識が要求され, 基幹システムからのレプリケーションを実現するためには多くの開発コストが予想される。

DataExtractor/IX では, このような問題を解決し, DMS のレプリケーションの作成やデータウェアハウスの構築を容易に実現するための機能を提供する。更新差分抽出を使用することによって, 既存システムを変更することなく, 抽出元データベース

の更新を抽出先データベースに動的に反映することが可能である。

本稿は、DataExtractor/IX を使用したレプリケーションの実現方法について解説するものである。2章では、データウェアハウスの構築において求められる基盤技術について説明する。3章では、DataExtractor/IX の機能と、それを使用したデータウェアハウス開発について述べる。4章及び5章では、DMS データベースのレプリケーションにおける更新差分抽出の技術的な利点と、更新差分抽出の効果的な利用方法や運用形態について述べる。

2. データウェアハウス構築に求められる基盤技術

2.1 ETL ツールとしての機能

通常、データウェアハウスの構成要素としては、データベース管理システム (DBMS)、ETL、及び情報分析ツール (OLAP、データ・マイニングツール) がある。ETL は、データ抽出 (Extract) / 変換 (Transformation) / ロード (Load) 処理の略称であり、これらの各機能は、データウェアハウス構築支援ツールの基本機能として要求されるものである。

ETL におけるデータ抽出は、基幹システムからのデータのバルク抽出^{*2} 処理であり、データ変換では、抽出したデータの統合・編集・加工・変換などの処理を行う。データロードは、加工されたデータを抽出先データベースへロードする処理である。図1では、データウェアハウス構築における ETL の役割を示している。DataExtractor/IX は、ETL ツールとして、CS シリーズ (OS 2200) 上の各種データソースからデータを抽出し、データの編集・加工、文字コード変換等を行い、オープンシステム (Windows/UNIX) 上の RDB へロードするための機能を提供する。

また、DataExtractor/IX では、ETL ツールとしてのデータ抽出に加えて、DMS データベースの更新差分をトランザクション単位で反映する更新差分抽出機能を提供する。更新差分抽出では UDS コントロールと連動して、オーディット上に採取したトランザクション・ログを収集・編集・加工した後、抽出先データベースに反映する。これによって、抽出先データウェアハウスは、常に基幹システム上の DMS データベースと同期を保つことが可能である。

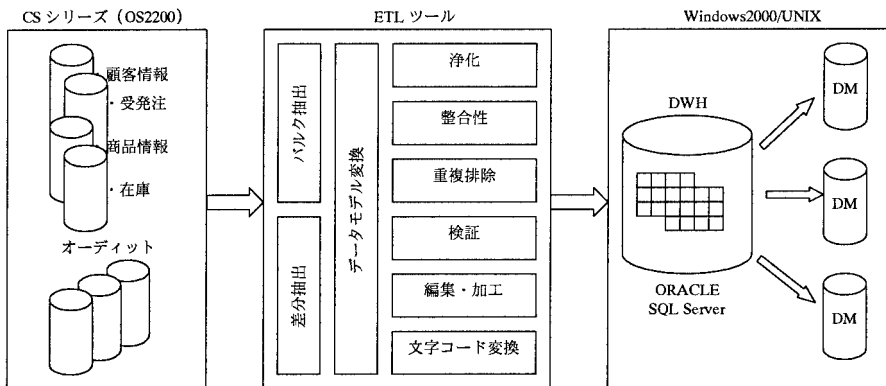


図 1 ETL の役割

2.2 レプリケーション機能

データのレプリケーションとは、データソースと同期のとれたデータの複製を、対象とするサーバ上に構築するための技術である。データベース製品におけるレプリケーションは、データウェアハウス (DWH) 構築^[2]やインターネット技術の発達を背景としたソリューション指向の技術であると捉えられる。

レプリケーションと従来型の分散データベース^[1]との大きな違いは、分散データベースがデータベースを静的に分割配置するのに対して、データベースの複製の作成や配置が自由な点にある。このため、レプリケーションを利用することで、Web 連携アプリケーションやOLAPなどのデータ分析アプリケーションを容易に実現することが可能である。レプリケーションには、以下のような利点がある^{[3][4]}。

- ・複製を分散させることにより、可用性が高まる
- ・複製を分散させることにより、アクセス集中を防止し効率を高める
- ・アプリケーションは複製のみをアクセスすればよく、ネットワーク負荷が生じない
- ・OLTP、OLAP など用途に応じて、データベースの複製を作成することが可能である

また、レプリケーションは、複製元のデータベースと同期を取る仕組みによって、基本的に以下のように分類される。

- ・静的レプリケーション
ある時点での複製元データベースの静的なコピーを作成する。
- ・動的レプリケーション
複製元データベースに加えられた変更を複製先データベースへ反映し、複製元と複製先との間で、常にデータベース情報の同期を取る。

データウェアハウス構築ツールとしてレプリケーションを使用するためには、複製の作成に加えて、ETL ツールとして必要なデータ変換機能が備わっていなければならない。DataExtractor/IX では、レプリケーションの基本的な機能に加えて、モデル変換、文字コード変換、データ加工などの ETL ツールとしての機能を提供する。DataExtractor/IX におけるデータ抽出機能、及び更新差分抽出機能は、ネットワーク型データベースからリレーショナル・データベースへの、異なるデータモデル間におけるレプリケーションを実現したものである。

2.3 抽出ツールの自動生成機能

データウェアハウス構築を効率的かつ低コストで行うための技術として、ETL (抽出、変換、ロード)を行うためのツールを自動的に生成するための機能が要求される。DataExtractor/IX では、ノンプログラミングでレプリケーションを可能にするための機能を提供する。使用者は、抽出先 RDB スキーマの設計を行い、抽出元 DMS スキーマとのマッピングを定義することで、専用の抽出プログラムを生成することが可能である。生成したツールは、直ちに実行することができ、抽出ツールのメンテナンスに関しても、ワークベンチを使用することによって、運用、管理、保守を容易に行うことができる。

また、抽出ツールが自動生成されることで、従来の開発プロセスに比較して作業工

程が大幅に短縮される。開発者はCSシリーズ (OS 2200) におけるソフトウェア開発の技術やネットワーク型データベースに関する特別なスキルから開放されると同時に、プログラムの設計・開発やデバッグが不要となる。

3. データウェアハウスの構築と鮮度維持

3.1 抽出元データソースと抽出先 RDB

DataExtractor/IX では、CSシリーズ (OS 2200) 上の DMS データベース、各種 COBOL ファイルをデータソースとし、オープンシステム上の RDB へ抽出したデータをロードする。抽出先 RDB としては、Oracle、Microsoft SQL Server をサポートする。DataExtractor/IX の抽出対象となるデータソースは以下のとおりである。

- ・ DMS データベース
- ・ COBOL ファイル (順編成ファイル、指標付き順編成ファイル、多重索引順編成ファイル)

抽出先 RDB は、以下のとおりである。

- ・ Oracle (Windows NT/2000, Sun Solaris, HP UX)
- ・ Microsoft SQL Server (Windows NT/2000)
- ・ CS シリーズ UDS RDMS 1100

DMS データベースの抽出については、初期抽出と更新差分抽出の二つの抽出形態を提供する。初期抽出は静的レプリケーションに相当し、ある時点におけるデータベースの静的な状態の複写であり、主にデータウェアハウス初期構築に用いられる。更新差分抽出は動的レプリケーションに相当し、DMS データベース上の時系列な更新を抽出データベース側へ動的に反映するものである。これは、構築後のデータウェアハウスを常に最新の状態に保つことを目的として使用される。

3.2 DataExtractor/IX のコンポーネント

DataExtractor/IX は、CSシリーズ (OS 2200) で稼働するコンポーネントとオープンシステム上で稼働するコンポーネントとから構成される。オープンシステム上のコンポーネントには、クライアント PC で稼働するワークベンチと、RDB サーバ上で稼働するツール類とがある。以下に、DataExtractor/IX で提供されるコンポーネントを示す (図 2)。

1) ワークベンチ

Windows 上で稼働する GUI ツールであり、RDB スキーマと DMS スキーマとのマッピングや抽出プログラムの自動生成を行う。

2) RDB コンポーネント

抽出先ホスト上の RDB のテーブル情報を抽出するためのテーブル情報抽出ツールや更新差分抽出のためのツール類 (ファイル転送エージェント、差分反映ツール) を含む。RDB コンポーネントは、RDB の種別に応じて Oracle コンポーネントと SQL Server コンポーネントとがある。

3) 自動抽出ツール

ホスト上の DMS 抽出プログラムの実行からサーバ上の RDB へのローディングまでを実行するツールである。ワークベンチで定義したテーブルマップ情報を

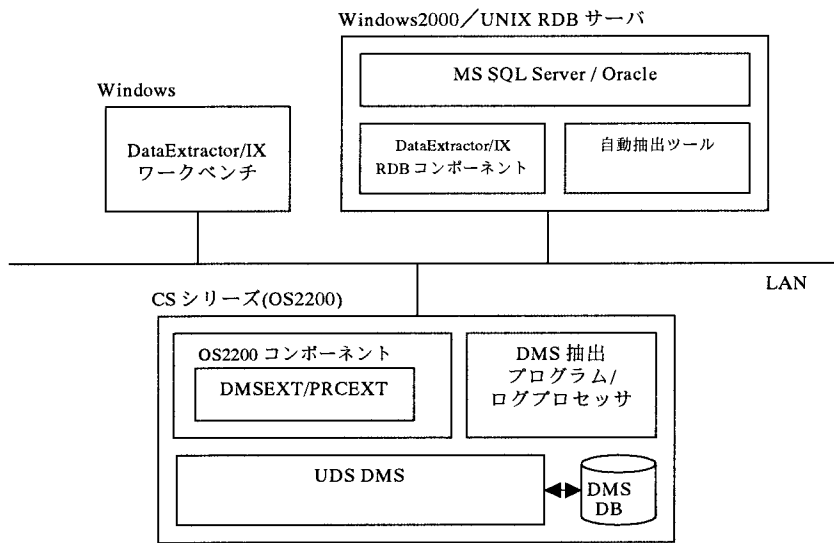


図 2 DataExtractor/IX のコンポーネント

元に生成され、各抽出先サーバへ転送される。

4) OS 2200 コンポーネント

CS シリーズ (OS 2200) 上の DMS スキーマ情報の抽出や PROC 情報の抽出を行う。抽出した情報は、スキーマ情報、使用者定義情報としてワークベンチに取り込まれる。DMS 抽出プログラムやログプロセッサの生成に必要な PROC やライブラリ等を含む。

5) DMS 抽出プログラム/ログプロセッサ

DMS 抽出プログラム、及びログプロセッサは、使用者のマッピング情報に特化した専用の抽出ツールである。プログラムソースがワークベンチによって生成され、ホストへ転送されてコンパイルされる。DMS 抽出プログラムは初期抽出用のツールであり、ログプロセッサは更新差分抽出のためのツールである。

3.3 抽出項目のマッピング

抽出項目のマッピングは、DataExtractor/IX の GUI ツールであるワークベンチを使用して行う。DMS の抽出対象レコードの項目と抽出先 RDB テーブルの項目との割り付けをテーブルマップ情報として定義する。ここで定義された情報は、初期抽出、更新差分抽出において抽出処理の基本単位となるものである。以下に、テーブルマップ情報の作成手順を示す。

1) 抽出先 RDB テーブル情報の取り込み

RDB コンポーネントのテーブル情報抽出ツールを使用して、抽出先 RDB のテーブル情報を抽出し、ワークベンチに取り込む。

2) DMS スキーマ情報/PROC 情報の取り込み

OS 2200 コンポーネントの DMSEXT、PRCEXT プロセッサを実行し、抽出対象 DMS スキーマの定義情報と PROC 定義情報を抽出しワークベンチに取り込む。

3) 使用者定義マップ情報とスキーマリンク情報の設定

レコード項目に対して再定義が必要な場合は、使用者定義マップ情報を編集して再定義情報を割り当てる。抽出対象とするレコードをスキーマリンク情報として設定する。レコードは線形のセット関係で定義されるものに限り、複数含めることができる。このとき、設定したレコードに対して抽出条件を設定し、抽出範囲を限定することができる。

4) テーブルマップ情報の作成

DMS のレコード項目と、SQL のテーブル項目の対応付けを行い、テーブルマップ情報を作成する。必要であれば、抽出データの形式、文字コード、データの加工などを定義する (図 3)。

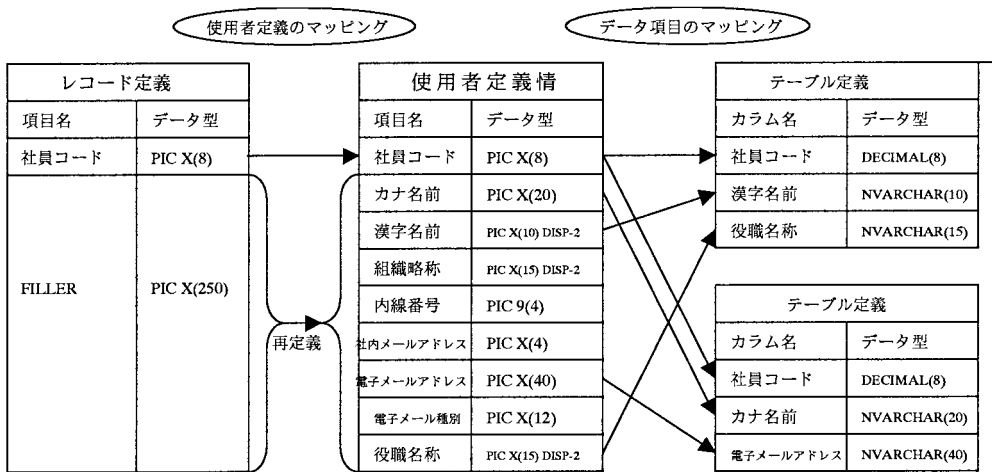


図 3 レコード項目とテーブル項目の対応付け

3.4 初期抽出の実行

ここでは初期抽出ツールによるデータベースの初期構築について説明する。初期抽出では、DMS データベースからデータを静的な状態で抽出し、サーバ上へ転送した後、RDB へのロードまでの処理を行う。使用者は、ワークベンチ上で定義したテーブルマップ情報から、DMS 抽出プログラムと自動抽出ツールを生成する。自動抽出ツールは、DMS 抽出プログラムの実行から RDB へのロードまでの一連の工程を、サーバ側から自動的に実行するためのツールである。図 4 では、自動抽出ツールの処理の流れを示している。自動抽出ツールが起動されると、CSシリーズ (OS 2200) 上の抽出プログラムを起動し、データ抽出を実行する (図 4 の①)。抽出されたデータは抽出実行ホスト上で受信され (図 4 の②)、RDB ベンダー固有のデータロードユーティリティを起動してデータのロードが行われる (図 4 の③)。

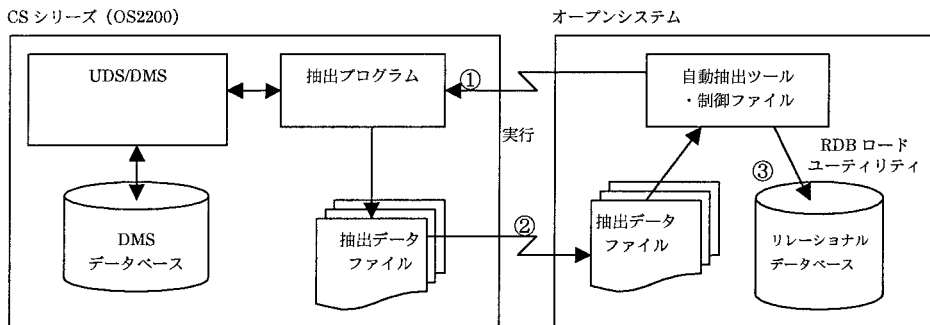


図 4 初期抽出の実行

3.5 更新差分抽出の実行

更新差分抽出は、DMS の差分データを抽出先データベースへ反映することによって、抽出先データの鮮度を維持するものである。更新差分抽出の処理は、DMS の処理フェーズと DataExtractor/IX の処理フェーズとに分かれている。DMS フェーズでは、UDS コントロールによるオーディットへの更新データの採取と、ログプリプロセッサによるオーディットの編集処理が行われる。DataExtractor/IX フェーズでは、プリプロセッサによって収集されたトランザクション単位のログの編集と、サーバ上の差分反映ツールによる抽出先 RDB への反映が行われる。

1) 更新ログの採取

DMS では、DCT (データ・キャプチャー・テーブル) に登録されたレコードの更新ログをトランザクション単位で、ログオーディットに採取する。採取された更新ログは、ログプリプロセッサによってトランザクション単位に収集され(ステージング処理)、スキーマ単位のイベントログファイルが生成される(図5の①)。

2) 差分データファイルの生成

イベントログファイルに書かれたログレコードは、ログプロセッサによって、テーブルマップ単位に編集・加工が行われ、差分データファイルが生成される(図5の②)。ログプロセッサは、指定された間隔で転送指示メッセージファイルを生成する。転送指示メッセージファイルは、抽出先サーバ上のファイル転送エージェントが、差分データファイルの処理を開始するためのトリガーとなる情報ファイルである。

3) レプリケーション処理の実行

ファイル転送エージェントは、転送指示メッセージファイルを定期的に検査し、ファイルが作成された場合は、差分データファイルの受信を行い、差分反映ツールを起動する。差分反映ツールは、差分データファイルから SQL スクリプトを生成し、RDB のコマンドラインユーティリティ (Oracle の SQL・Plus や SQL Server の osql) によって差分データの反映を行う(図5の③)。

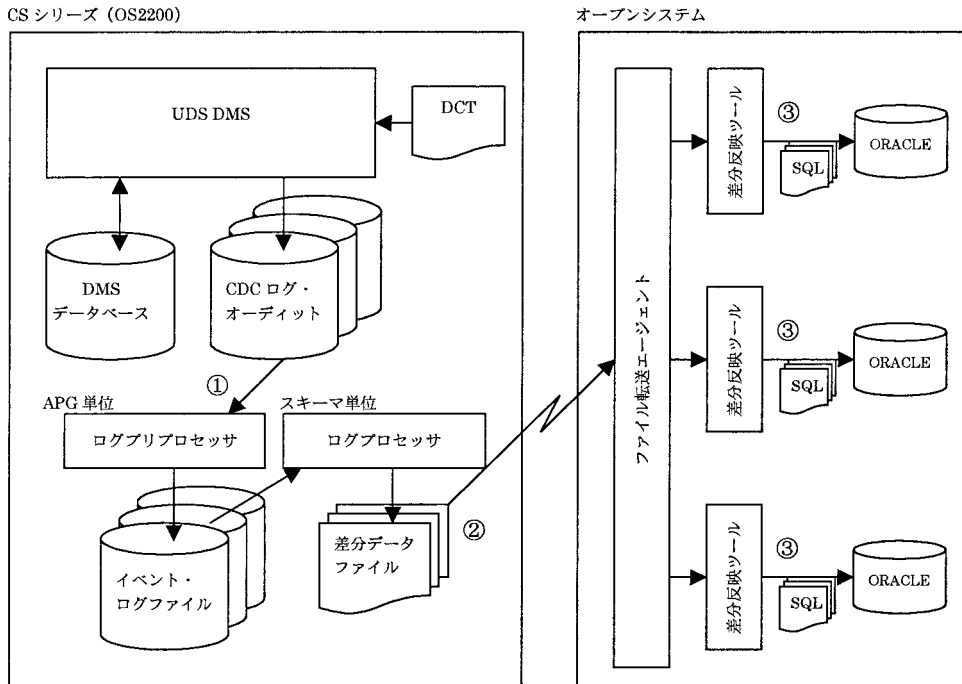


図 5 更新差分抽出の実行

4. DataExtractor/IX の機能と特長

4.1 UDS コントロールによる効率的な差分ログの収集

オーデット・トレイルは、更新ログの抽出先としては最適な媒体である。更新ログは、オーデット・トレイル・テープまたはオーデット・トレイル・ディスクにオーデット・レコードとして書かれる。システム障害時でも更新ログは決して消滅することはない。

また、XTC UDS^{*3} 環境下でも、アプリケーション・グループの CSN (Commit Sequence Number)^{*4} が割り当てられるため、各ホストで採取された更新ログの順序性を保つことができる。

しかし、更新ログの採取に伴いオーデット量が増大し、システム障害発生時のデータベースのリカバリに時間がかかることが予想されるため、その対策として、主オーデット以外に更新ログ専用のオーデットを用意することができる。これは、UDS が稼働しているアプリケーション・グループ以外のアプリケーション・グループのオーデットを更新ログ専用のオーデットとして使用するものである。

4.2 ホスト間の通信形態

更新差分抽出における通信形態は、ホスト上でログプロセッサが生成したファイルを、クライアントであるファイル転送エージェントがサーバ側から受信するプル型システムである。このため、ホスト/サーバ間で情報を共有する必要がなく、高い独立性が保障される。更新差分抽出の中断や再開などのオペレーションは、ホスト側からでもオープンシステム (Windows/UNIX) 側からでも容易に行うことが可能である。これは、障害発生時の再開、及び回復処理においても有効なものである。

ログプロセッサでは、定義されたテーブルマップごとに、イベントログファイル上のDMSの更新コマンド単位でログパケットを処理する。CIP ログ^{*5}を検出した場合は、差分データファイルにコミット（COMMIT）コマンドを追加する。また、ログプロセッサのインターバル・パラメタで指定された間隔で差分データファイルをクローズし、転送指示メッセージファイルを作成して転送可能な状態とする。ログプロセッサとファイル転送エージェントの全体的な処理を示す（図6）。

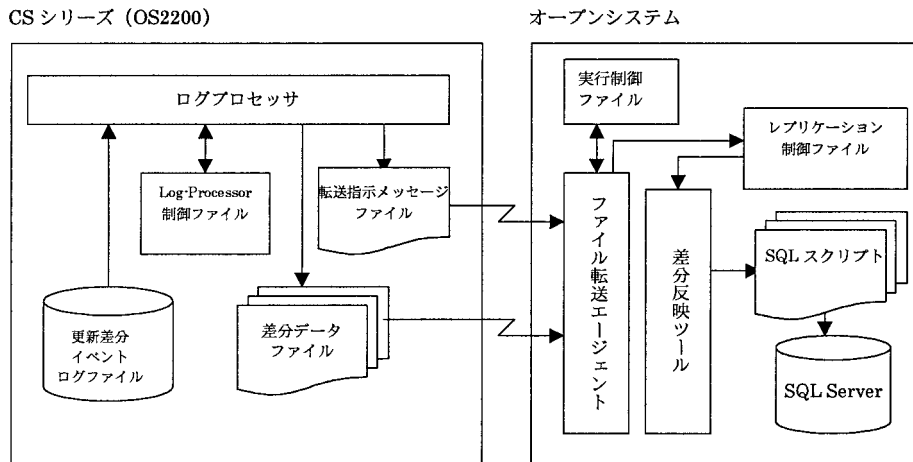


図6 ログプロセッサとファイル転送エージェントの処理

ファイル転送エージェントでは、テーブルマップ単位の差分反映処理を、マルチスレッドによって並列に実行している。これらは、Windows または UNIX のマルチスレッドによって実行され、効率のよい差分反映を実現している。ファイル転送エージェントは、転送指示メッセージファイルをトリガーとして、対応するテーブルマップの差分データファイルを受信し、RDB への反映までを行う。

4.3 耐障害性の保証

更新差分抽出は、CS シリーズ (OS 2200) ホスト、PC/UNIX、及びそれらを接続しているネットワーク上で稼働するものである。このため、DataExtractor/IX 更新差分抽出では、障害の発生地点（表1）に関わらず、容易に復旧できる機能を実現している。

ログプロセッサでは、更新コマンドのログを一件処理するタイミングで制御ファイルに情報を記録している。また同時に、ファイル転送エージェント側でも、ファイル

表1 障害発生のポイントと主な原因

障害箇所	主な原因
CS シリーズ(OS2200)側の障害	CS シリーズ(OS2200)ホスト障害、UDS DMS アボート、APG アボート
RDB サーバ側の障害 (PC/UNIX)	PC/UNIX サーバ障害、RDB 障害
通信障害	CS シリーズ(OS2200)ホスト/サーバ間の通信回線障害

受信処理/差分反映処理サイクルごとの処理状況を記録している。このような独立した情報管理によって、更新差分抽出の中断と再開を、任意の時点でホスト側またはPC側のどちらからでも自由に行うことができる。

これによって更新差分抽出では、障害の発生地点に関わらず、ホスト/サーバ間の同期を意識せずに処理を再開することが可能である。

4.4 SANARENA ディスクによるデータ通信効率の向上

DataExtractor/IX では、UHMDE (高速データ交換ソフトウェア)⁶⁾対応を行っている。UHMDE は、SANARENA ディスク上の共有ボリュームを使用して、CSシリーズ (OS 2200) とオープンシステム (UNIX/Windows) 間におけるデータ交換機能を提供するものである。UHMDE によるデータ交換処理は、ネットワークを経由せず直接ディスクを介して行われる。そのため、ネットワークによる通信負荷がかからず、高速なデータ交換が実現される。DataExtractor/IX は、UHMDE を使用することによって、データ転送にかかる処理時間を短縮し、抽出からデータロードまでの処理を効率的に行うことが可能である (図7)。

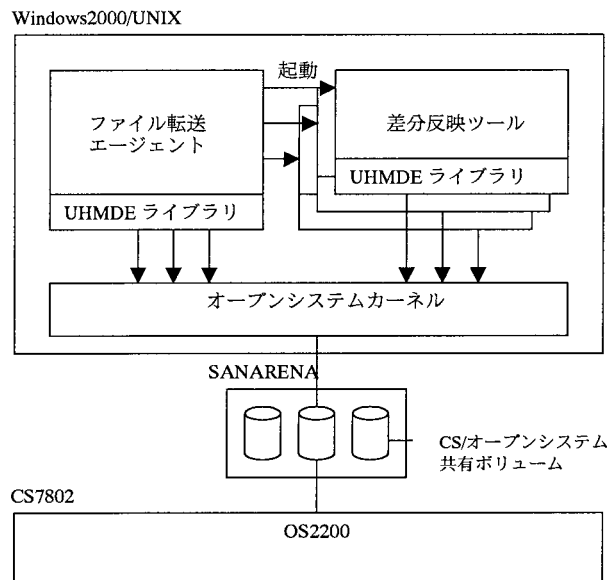


図7 UHMDEを使用した更新差分抽出

5. 更新差分抽出の運用

5.1 差分の反映形態

DataExtractor/IXの更新差分反映は、テーブルマップ単位に行われるが、テーブルマップごとの抽出先RDBサーバはそれぞれ異なってもよい。このため抽出されたデータを一つの抽出先サーバ上に集めることも、それぞれ別の抽出先サーバ上に分散させて抽出することも可能である。

抽出先ホストの選択は、DCT情報の生成時に指定することが可能であり、運用の形態に合わせて自由に変更することができる。このように業務プロセスや用途に応じ

て、抽出形態を柔軟にチューニングすることができることは、DataExtractor/IX の大きな利点の一つである。図8は、基幹データベース上の各支店ごとの情報を、それぞれの支店ごとのサーバに分散して抽出する例である。

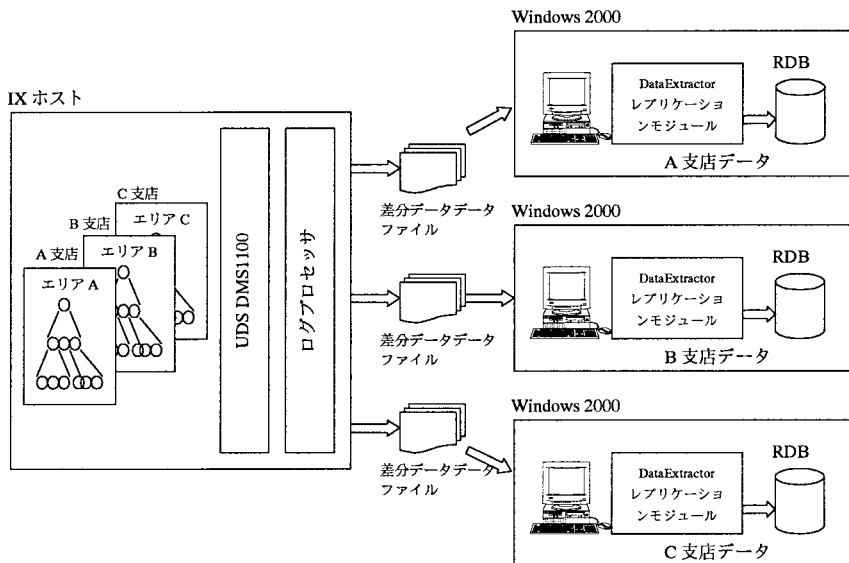


図 8 抽出先ホストが分散している例

5.2 リアルタイム処理とバッチ処理

更新データのターゲットホスト上への反映には、ホスト上のデータベースと同期を取って逐次的に差分反映を行うリアルタイム処理方式と、日次、週次の周期で定期的にバッチ処理的に差分反映を行うバッチ処理方式とがある。

リアルタイム処理は、ホスト側のログプリプロセッサ、ログプロセッサの常駐化とオープンシステム側でのファイル転送エージェントの常駐化によって実行することができる。リアルタイム処理においては、更新ログの処理時間とデータ転送による遅延が生じることを考慮すべきである。バッチ処理方式では、周期として日次単位、週次単位、月次単位などが考えられるが、中間データの保存形態によって、運用形態が異なる。例えば、ログ用オーディットを保存する場合は、ログプリプロセッサ・ログプロセッサをバッチ的に運用する方法となり、イベントログファイルを保存する形態では、ログプロセッサをバッチ的に運用する方法となる。また、差分データファイルを保存する形態では、抽出先ホスト上でのファイル転送エージェントをバッチ的に実行することが可能である。

6. おわりに

本稿では、DataExtractor/IX の機能とデータウェアハウス構築やレプリケーション技術における位置付けを概説し、更新差分抽出で提供されるデータ抽出技術とその特長について説明した。CS シリーズ (OS 2200) の基幹データベースとしての DMS に対するレプリケーション技術は、IT 適用の推進と従来のデータウェアハウジング

の両面から、今後さらに重要性が高まるものと予想される。しかしながら、DMSを基幹データベースとするユーザの運用形態は多様であり、顧客の業務プロセスごとに異なるレプリケーションのパターンが存在していることを考えると、DMSとオープンシステムとのデータ連携の実現は容易なことではない。DataExtractor/IXでは、システムのさまざまな運用形態に適用可能な機能を備えており、多様な要求に対する柔軟なソリューションを提供する。

-
- * 1 UDS DMS 1100 : OS 2200 上の汎用データベース管理システム (DBMS) であり、UDS (Universal Data System) 上で DMS 1100 (ネットワーク型データベース)、RDMS 1100 (リレーショナル・データベース)、SFS 1100 (階層型データベース) が稼働する。UDS コントロールは、データモデル部分を除くデータベース管理システム部分を指す。
 - * 2 バルク抽出 : 大容量抽出処理のこと
 - * 3 XTC UDS : XPC (拡張トランザクション処理アーキテクチャ) に対応した UDS コントロール。
 - * 4 CSN : XPC によってオーディット・ブロックの順番に割り当てられる番号。
 - * 5 CIP ログ : DMS の DEPART、FREE コマンドに対して採取されるログであり、ログプリプロセッサが、オーディット上の CIP (Commit in Progress) レコードから作成する。

- 参考文献** [1] D. Ensor, I. Stevenson, Oracle システム設計, オライリー・ジャパン, 1997, pp 317-353.
- [2] H. Baer, Oracle 9i での ETL 処理, Oracle ホワイトペーパー, Oracle Corporation, 2001.
- [3] Microsoft, レプリケーションアーキテクチャ, SQL Server Books Online, Microsoft Corporation, 2002.
- [4] W. Creekbaum, R. Urbano, Oracle 8i レプリケーション・ガイド, リリース 8.1, Oracle Corporation, 2000.
- [5] 『ClearPath OS 2200 DataExtractor/IX 解説書レベル 3』, 日本ユニシス株式会社, 2002.
- [6] 『UNISYS SANARENA 2000 シリーズ UHMDE (高速データ交換ソフトウェア) 解説書レベル 1』, 日本ユニシス株式会社, 2002.

執筆者紹介 実藤 英昭 (Hideaki Jitto)

1985 年早稲田大学商学部卒業。同年日本ユニシス(株)入社。HMP IX シリーズデータベース管理システムの開発保守に従事。現在プロダクトサービス部 IX ソフトウェア室所属。

中山 陽太郎 (Yotaro Nakayama)

1988 年東京学芸大学大学院教育学修士課程修了。同年日本ユニシス(株)入社。HMP IX シリーズデータベース管理システムの開発保守に従事。現在プロダクトサービス部 IX ソフトウェア室所属。