

効率と拡張性を両立するシステム間データ連携方式の提案

楓 仁志† 石井 篤† 高山 茂伸†

† 三菱電機株式会社 情報技術総合研究所 〒247-8501 神奈川県鎌倉市大船 5-1-1

E-mail: † {Kaede.Satoshi@cw, Ishii.Atsumi@bp, Takayama.Shigenobu@db}.MitsubishiElectric.co.jp

あらまし 近年、企業情報システムは、さまざまなシステムを適材適所で連携して利用することが求められている。情報システムを連携する方式において、システム間のデータ連携を個別に実装する方式と、基盤となるシステムを導入することでデータ連携を一元化して実施する仕組みがある。筆者らは、データ連携を一元化する仕組みにおいて、企業情報システム間のデータ連携に必要と考えるデータ連携処理の証跡管理、複数のデータ連携処理の重複排除による高速化、連携対象システムの拡張性に特化したデータ連携システムの実現方式を提案し、プロトタイプ実装を報告する。

キーワード データ連携, データ統合

A Method of Interconnected Information Systems focused on the processing efficiency and system extensibility

Satoshi Kaede † Atsumi Ishii † Shigenobu Takayama †

† Mitsubishi Electric Corporation, 5-1-1 Ofuna, Kamakura, Kanagawa 247-8501, Japan

E-mail: † {Kaede.Satoshi@cw, Ishii.Atsumi@bp, Takayama.Shigenobu@db}.MitsubishiElectric.co.jp

Abstract In this paper, we propose an interconnected information systems method for efficiency of throughput and extension for appending other information system. Recently, it is necessary to construct information system which provides requirement method by integrated existing information systems. We study this problem in system interconnection, we proposed a data integration platform system which using data trail management, efficiency method of data interconnection processes, and system extension method. Finally, we present a data integration system prototype with proposed method in this paper.

Keyword Data Integration, Information System Interconnection.

1. はじめに

1.1. 企業情報システムの構築

従来、企業情報システムは各業務に応じた情報システムを構築し、各システム間でデータ連携が必要な際には、バッチ処理などのデータ転送用アプリケーションを開発することで複数の情報システムを連携させてきた。

昨今の企業情報システムは、組織改正への対応や事業体の変化、買収合併による統廃合など事業の形態に柔軟に対応することが求められている。個々の情報システムのデータ連携を実施する都度、連携用アプリケーションを開発するのではなく、既存情報システムや外部の情報システムと柔軟に連携する仕組みが必要である。

1.2. 疎結合による情報システムの構築

近年、クラウドコンピューティングや SaaS というキーワードにおいて、複数の情報システムを連携させて新たな価値を生み出す、もしくは低コストで業務要

件を満たすといった情報システム構築の形態がある。情報システムを連携させるという言葉の真意は、情報システム間で参照するデータを共有することであり、同一の情報を複数の情報システムが利用することを意味する。

例えば企業内プライベートクラウドが構築され、人事や経理といった業務システムがクラウド環境へ移行し、事業場所を跨って利用される。このような場合においても、特定の情報システムで発生したデータを他の情報システムで利用したいという要求仕様が変更することはない。そこで情報システムを密に結合させる従来の方式ではなく、データ連携の実施を一元化する基盤システムを導入する。これにより、複数の情報システムを効率的に連携させることを目的に筆者らは情報システムが備えるデータベース間でデータを連携する基盤システムの一実現方式を提案する。

1.3. 本研究の目的

情報システムの構成を 5 階層で表現した場合、例えば図 1 のように定義できる。Web システムのマッシュ

アップと呼ばれる技術は、プレゼンテーション層によるシステム連携と言える。バッチプログラムによるデータ転送などはアプリケーション・ロジック層によるシステム連携を実現する方式と言えるであろう。



図 1 情報システムアーキテクチャ

アプリケーション・ロジック層によりシステムを連携させた場合、業務要件の変更による情報システムの改修に対応するためには連携対象となる情報システムに対する連携処理の停止/再開や、データ仕様の変更といった改修が発生することとなる。そこで筆者らは、これら課題を解決するために情報システムの最下位層に位置するデータ層を対象としたシステム間連携を実現する手法について開発を行なっている[1]

2. データ連携システムの課題

クラウドなど社外情報システムと企業内情報システムが連携する場合、各情報システムの管理会社は異なる。この場合、データ連携システムで発生した障害の影響を連携対象となる情報システムまで及ぼすことは回避しなければならない。例えばデータ連携システムが異常停止した場合に、連携元システムから対象データを再取得するといった事態は回避しなければならない。

また、近年 CDC (Change Data Capture) [2]のようにデータの発生元システムに更新を検知する仕組みを組み込むことで準リアルタイムに情報システムで発生したデータを他の情報システムで利用することが求められている。CDCは企業情報システムのオンライン処理中に稼動するものである。そのため、特定の連携先システムが増加した際に、連携元システムで稼動するCDCなど運用中の既存のデータ連携処理を停止し、設定情報を変更したのち再開させるといったメンテナンス作業やデータ連携処理の遅延が発生する事態も回避しなければならない。

3. 提案手法

3.1. 効率と拡張性を備えたデータ連携システム

同一のデータを複数の情報システムが利用する際、各情報システムの管理体制が異なるという前提におい

て筆者らは、データ連携システムにおける障害発生時や、データ連携の対象となる情報システム拡張時の独立性に着目する。情報システムの独立性を担保したデータ連携システムを実現する方式として筆者らは、企業情報システム間のデータ連携において以下の機能が必要と定義する。

機能 I : 連携対象データの証跡管理機能

機能 II : 複数のデータ連携処理の重複排除による高速化機能

機能 III : 連携対象システムの拡張機能

3.2. 機能 I 連携対象データの証跡管理機能

各情報システムの管理体制が異なることから、例えば、情報システムを連携させるデータ連携システムに障害が発生した際にその影響を連携元システムや連携先システムに及ぼすことは回避しなければならない。このような連携対象システムとデータ連携システムの独立性を担保するために、連携元システムからデータ連携システムが取得したデータを永続化させる証跡管理が必要である。

そこでまず、データ連携システムは共有ディスク装置などを用いて、連携元情報システムから取得した連携対象データを永続的に保持する。次に本連携データの証跡管理機能をもとに、異常が発生した際には待機系サーバに切替え、証跡管理機能に格納された処理対象データを用いて、自動的に連携先情報システムに配信するといった仕組みが実現可能である。

3.3. 機能 II データ連携の高速化機能

連携対象データの証跡管理を実施するためには、連携元システムからデータ連携システムがデータを受信した都度、永続化処理が必要となる。そのため、証跡管理を実施することによる性能劣化が懸念される。

データベースの分野において、複数のデータベース間でデータを連携する手法として、ETL (Extract Transform Load) などがある。ETLはDWH (Data Warehouse) を構築する際にデータベースからデータを移行するために利用され、またDWHから業務分析対象に特化した情報だけを抽出したデータマートを形成する際などに利用されている。従来のETL手法は、図2のように個々の連携対象システムのDBの項目ごとの対応関係(以降、DB対応関係と記す)を生成し、データ連携システムの内部処理はDB対応関係の設定と同様に、サーバ内のプロセスなどは個々の連携ごとに処理単位を生成する手法が取られてきたと考える。これは、バッチ処理を前提とした開発経緯に寄るものが大きいと考える。

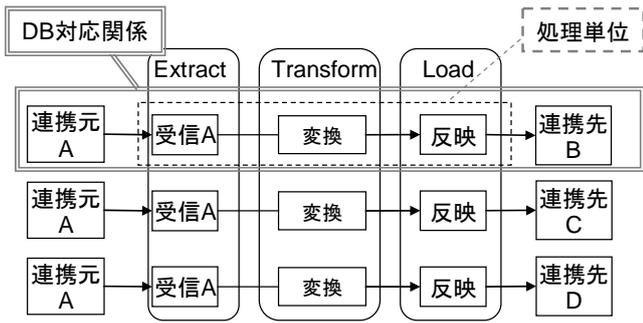


図 2 従来の設定情報によるデータ連携

しかし、CDC など不定期に発生するデータをリアルタイムに連携する際、更新データが発生するたびに逐一個々のデータ連携バッチを起動し、証跡を管理するためのデータ永続化処理を実施することは効率的ではない。そこで筆者らは、データ連携処理の証跡管理機能を実現する際に発生するデータ連携システムの処理性能劣化に対して、文献[3]、[4]のように情報システムの DB 対応関係の重複を排除する手法を開発している。本手法は、情報システム毎に定義した複数の DB 対応関係を入力とし、図 3 に示すように同一データを利用する情報システムが複数ある場合や、同一の変換処理によって目的とするデータを生成する場合の処理の重複を排除するものである。

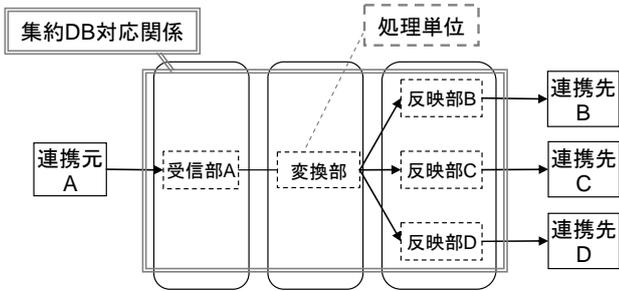


図 3 集約 DB 対応関係によるデータ連携

集約された DB 対応関係に基づき動作するデータ連携システムの処理単位は、複数の連携対象システム間で共有する構成をとる。ETL による従来手法では、連携先システム数が増加すると、都度連携元システムからデータを受信することとなり、データ連携の証跡を管理するための永続化処理が重複する。しかし、本手法を用いることでデータ連携の証跡を管理し、かつ同一連携元システムからのデータ受信の重複を排除することで連携対象システムの増加に伴う処理性能の劣化を抑制する。

3.4. 機能Ⅲ 対象システムの拡張機能

本来情報システムはその業務形態に応じて継続的に形態を変化するものである。特に SaaS など外部の情報システムと連携する場合は、必要に応じて連携先の情報システムを変更可能とする機能が必須と考える。

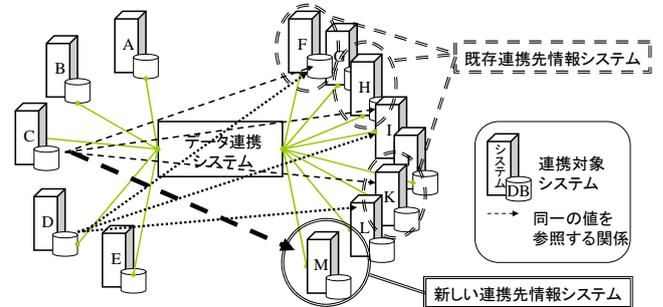


図 4 連携対象システムの拡張

図 4 に示すように、例えば既に稼動している情報システム間のデータ連携が存在する状態で、新たに連携先システムをデータ連携システムに追加する場合、既存のデータ連携処理を停止することは許されない。従来の ETL のように個々の情報システムをピア to ピアで定義した DB 対応関係を元に動作する場合には、新たな DB 対応関係の設定を追加することになるため、機能 I によるデータ永続化によって性能劣化が発生するため要件を満たさない。また、既存のデータ連携を実施する DB 対応関係に対して連携先情報システムを追加する場合には、既存データ連携処理の停止・再起動が必要となるため要件を満たすことができない。

そこで、本書では、以下に記載する手段により既存のデータ連携処理を停止させずに、新しい連携先情報システムを追加する。本方式は、新しい連携先システムの設定情報が追加された DB 対応関係に基づくデータ連携システムの内部処理構成を変更する時間を最小化し、既存データ連携処理への影響を抑制するものである。データ連携システムの内部処理構成の変更に要する時間を既存データ連携の処理性能に対して細微な遅延時間の発生に留めることで、効率と拡張性を備えたデータ連携システムを実現する。図 5 に処理概念図を示し、以下 Step①～Step④にて処理概要を記す。

Step①：システム M を連携対象に追加する際、システム C とシステム M の DB が持つ項目の新しい対応関係を設定し、新しい DB 対応関係を追加する。提案方式では、機能 II による設定情報解析処理を行ない、DB 対応関係の重複排除処理を行なう。このとき、実行中の集約された DB 対応関係と新しい DB 対応関係の差分を抽出する。

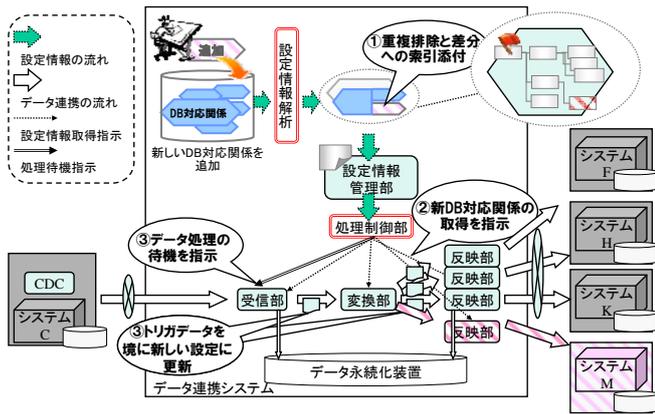


図 5 システム拡張の処理概念図

集約された DB 対応関係の差分を特定する索引を新規 DB 対応関係に付与する。本索引は、DB 対応関係に基づくデータ連携を処理する各処理部が構成変更に必要な負荷を考慮してソートする。

Step②：各処理部は DB 対応関係に基づいて処理を実行する。実行中の処理部の DB 対応関係を更新すると、データ連携処理を中断する必要がある。本方式では、各処理部は処理実行時に参照する DB 対応関係を複数保持する機構を備え、DB 対応関係の更新処理の事前に新 DB 対応関係を取得する。これにより、新しい DB 対応関係を各処理部に配布する処理は、新しい DB 対応関係への構成変更時間から除外することが可能となる。

Step③：新しい DB 対応関係を反映する時刻の到達と同時に、データ連携システムの受信部がデータ連携処理を一次待機する。これは同一の情報システムを連携元とするデータ連携処理の DB 対応関係の更新タイミングを一様に制御し、DB 対応関係の設定処理と処理対象データの不整合が発生することを回避する。新しい DB 対応関係を反映する時刻の到達と同時に、処理制御部がデータ連携システムの受信部に対して更新タイミングを通知するトリガーデータを挿入する。各処理部はトリガーデータを取得したタイミングで DB 対応関係の入れ替え行なう。

Step④：Step③によってトリガーデータが各処理部に伝達されることで、各処理部は Step②で事前に取得しておいた新しい DB 対応関係に切替え、必要な処理構成の変更を実施する。各処理部は、自身の処理構成変更が完了したのち状態管理を運用開始待ちに変更する。処理制御部は状態管理テーブルを監視し、Step①によって生成された索引に示された新しい DB 対応関係によって変更が

必要な全ての処理部の構成変更が完了したことを確認し、受信部のデータ処理一次待機を解除する。

以上の処理により新しい連携先情報システムが追加されたことをデータ連携システムの内部処理構成の変更処理に留め、かつ DB 対応関係の更新によって既存データ連携処理を実施する情報システムへのデータ連携処理の遅延時間を最小限に抑制する。これにより連携元システムとの通信再構築や、連携元システム上で稼動する CDC の再起動などは必要とせず、新規連携対象システムを追加することが可能となる。

4. プロトタイプ

本提案方式を適用したデータ連携システムのプロトタイプを構築した。図 6 に本プロトタイプのシステム構成を記載する。本プロトタイプは、連携元システムのデータベースの更新を検知する手段に商用の CDC 製品を適用した。また、連携先システムのデータベースに商用 DBMS を利用し、DBMS とのインタフェースは ODBC を用いて実現した。

連携元システムにて稼動する商用 CDC 製品は、データベースの実行ログを監視することでデータベースの更新を検知する機能を提供するものであり、インタフェースを実装することで、他システムとの通信処理を実装可能である。

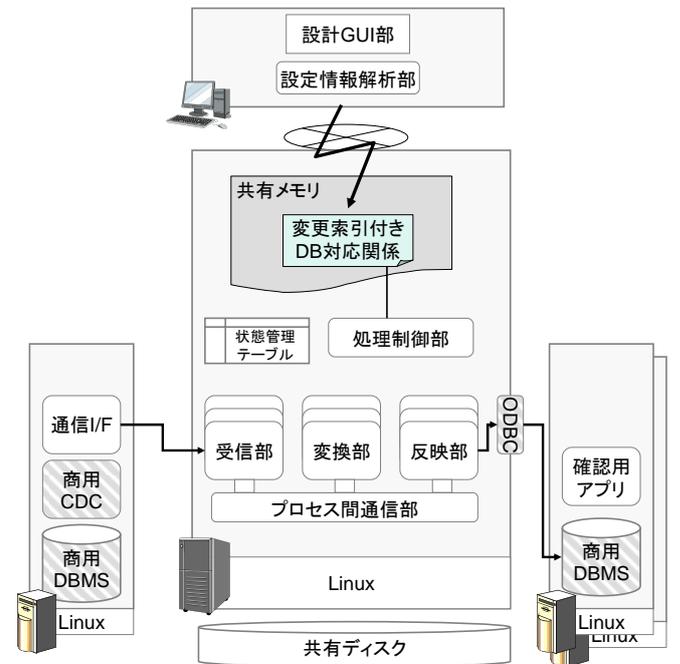


図 6 プロトタイプの構成図

本プロトタイプを用いて基礎性能を評価した。本評価は新しい連携先情報システムを追加する際に発生す

る既存データ連携システムへの遅延時間が、情報システム間のデータ連携の処理性能要件に及ぼす影響を確認することを目的とする。結果、連携先システムを追加した際に発生する既存データ連携処理性能要件に対する遅延割合は、1%以下と細微なものであることが確認された。

5. まとめ

本書では、クラウド環境などにおいて今後、益々増加すると予想される情報システムの連携を実現する一方式として、データ連携処理の証跡管理、データ連携処理の重複排除による高速化、設定情報への変更差分索引に基づく内部処理系の更新によるシステム拡張機能を備えたデータ連携システムを提案した。

今後は、本提案手法を実装したプロトタイプを基に連携対象システム数を増加させた際のスケールアウト機能、障害復旧機能の開発を行なう予定である。

参 考 文 献

- [1] 楓 仁志, 高山 茂伸, 菅野 幹人, 複数データソース間における効率的なデータ連携処理方式の提案, 情報処理学会第73回全国大会, pp.529-531, 2011
- [2] JinGang Shi, YuBin Bao, FangLing Leng, Ge Yu. Study on Log-Based Change Data Capture and Handling Mechanism in Real-Time DataWarehouse. ICCSSE 2008
- [3] 新井 嘉章, 楓 仁志, 石井 篤, 高山 茂伸, 大規模データ連携処理の実現に向けた実行時設計情報の管理方式, 情報処理学会第74回全国大会, 予稿集, 2012
- [4] 細田 聖人, 楓 仁志, 石井 篤, 高山 茂伸, 大規模データ連携処理の実現に向けた性能検討, 情報処理学会第74回全国大会, 予稿集, 2012