

事例

分散データベースの構築とその運用[†]

—総合建設業における全国展開と構築におけるポイント—

小林 正美^{††}

1. はじめに

当社では、平成3年の新社屋移転を契機に、各業務の仕組みを抜本的に見直し、コンピュータ資源を有効に活用できないか検討を進めてきた。

当社は総合建設業であり、全国の主要都市14カ所に支店が、県庁所在地をはじめとする約130カ所に営業所がある。また、建築や土木工事の施工管理をする作業所が約2千カ所あり、製造業の工場にあたる。これらの事業所における業務の特徴は、営業所や作業所といった生産拠点が遠隔地に散在し、そこで管理業務が主体となった単品生産が行われていることである。したがって、クライアント・サーバ（以降C/S）の分散型システムが業務形態にあった環境といえる。他方、オープンシステム技術は、従来のホスト中心の集中型システムからC/Sの分散型システムへの移行により、情報の有効利用や共有活用を可能とした。そこでパソコンによるLAN環境へ移行することとし2千台以上のクライアントが利用するためのデータベースを全国14支店に配置し、分散データベースの整合性をとる仕組みを構築した。

2. 分散型システム移行の背景と目的

2.1 移行の背景

従来からのコンピュータ環境は、「ホスト系」「オフコン系」「パソコン系」と、各々がほとんど独立した環境であった。

ホストでは、本社や支店のスタッフ向けのものが中心で、定型のデータ照会や帳票出力が多く、

ユーザが自由にデータを扱うことは難しい。

また、支店には分散機としてオフコンが導入され10年近く利用してきたが、4GL等の不足からシステムを開発しにくいし、OAツールも少ないので、手軽な利用環境ではない。

一方、PCはスタンドアロンだが、表計算、ワープロやグラフ等のOAソフトがあり、利用者がデータを加工できるので作業所まで含めた範囲で導入が盛んになってきた。

この状況を情報化投資と効果として捉えてみると、同じ程度の処理を行うオフコンとパソコンに対して二重投資を招いていると考えられるし、システム利用のためには各環境ごとにデータを入力する必要があり「データの二重入力」が発生し、非効率的な状況であった。もちろん、データを相互の環境でコピーすることは可能だが、必要なデータごとにファイル転送プログラムを作成しなくてはならないし、それを運用することはかなり複雑である。

一般のユーザが情報を活用する場合、どのコンピュータ環境かということは問題ではなく、必要な情報が、必要なタイミングで、自分の使いやすい道具で自由に取り出せるということが重要である。ユーザに対してこの環境を提供しやすいのは自由度の大きいパソコンである。ここ数年のパソコンの性能向上は目覚ましく、かなりの規模の処理が行えるし、GUIによる操作の理解しやすい使いやすいソフトが提供されてきている。

これらの状況から、全社的に統一された共通の利用環境を構築していく必要があった。

2.2 分散データベースの必要性

作業所のユーザは工事の竣工にともない1年から2年程度のサイクルで別の作業所へ移動する。

[†] Establishment and Utilization of Distributed Relational Data Base - The Keypoints of the Establishment and Utilization at General Contractor by Masami KOBAYASHI (Information Systems Department, Shimizu Corp.).

^{††}清水建設(株)情報システム部

このため全社のネットワークを構築するには柔軟性のあるシステム構成である C / S 環境が効果的である。これにより本社、支店、営業所および作業所といった当社のすべての事業所において、その規模の大小を問わず、同一のコンピュータ利用環境を提供していくことができる。そして、この共通基盤の上で統一されたアプリケーション（以降アプリ）を再構築していくことが可能となる。

アプリの特徴としては、個々の事業所で完結するものや、相互に情報を伝達し集約したり自分の管理情報と組み合わせる処理するといったものがあり、これらのデータ処理を行うためには拠点となる支店にデータベースが必要不可欠である。そこで、日常業務においてのデータの共用範囲を設定し、分散された DB を統合管理できる環境を整備していくことが重要である。

3. 分散データベースの概要

3.1 DB サーバ整備の基本的な考え方

オフコン系およびホスト系のシステムを、順次 C / S へ移行する計画であるが、これらの顧客管理や工事情報などのアプリは LAN 上に配置されたデータベース・サーバ（以降 DB サーバ）を利用することになる。

一方、ホストは C / S 環境における大きな 1 つのサーバという捉え方で、全社分の大量データを扱う処理が中心になっていくと考えている。

そこで、C / S における DB サーバの構成は以下の考え方で整備を進めることとした。

- (1) DB サーバの設置は部門や支店単位とし、ユーザは自部門のサーバを利用する。
- (2) DB サーバの増設は業務アプリやアクセスする利用者の増加を予測しながら実施する。
- (3) アプリのデータ更新は、部門 DB サーバに対して行う。
- (4) 全社データを扱う利用者（アプリ）のための DB サーバを設置する。
- (5) ホストのアプリへ C / S のデータが連携できる仕組みを構築する。
- (6) 個人利用のための DB は、クライアントまたはファイル・サーバへ配置し、利用者自身が運用管理を行う。

次に、この考え方に基づいた DB サーバの分類を示す。

(1) 部門 DB サーバ

部門で管理する範囲の DB を配置し、アプリから更新する。

(2) セントラル DB サーバ

部門のアプリの中でも他部門のデータ参照を必要とする場合がある。そこで参照用の全社分データを 1 台の DB サーバへ配置し、部門 DB との整合性を維持する。このサーバをセントラル DB サーバと呼ぶ。

(3) パーソナル DB

データベースソフトなどを利用するときのスタンドアロンの DB である。

3.2 分散 DB 構築の実施ステップ

従来の環境の中に新たに C / S 環境を構築し業務システムを一斉に移行することはできない。したがって、段階的にアプリの移行を進めていく。

現状のオフコン系システムは、部門間で類似しているが同一ではないという状況にある。これらを整理し本質的なニーズに基づいてアプリを再構築していくには複数の業務で共用されているマスタが C / S の DB に必要である。そこで、**図-1**「分散 DB 構築のステップ」に示したように移行を進めていくこととした。

まず、ステップ 1 を「分散 DB マスタの参照」段階とし、ホストで管理されているマスタ DB を部門 DB サーバの「共用マスタ DB」へ入れ部門内で参照利用が可能とした。

部門で利用するアプリからは部門独自に更新される個別 DB とこの共用マスタ DB を組み合わせる利用できるようにしている。また、個別 DB のデータは、必要があればホストへアップロードして従来の入力処理の代わりにすることができるようにした。これらの DB の整合性を維持する機能はバックエンドの仕組みであり、すべての処理が自動的に行えるように検討を進めてきた。

次のステップ 2 は「分散 DB のマスタの更新」段階と考えており、このステップの実現で共用マスタ DB の更新の流れを「DB サーバからホスト」へと切り替えることができる。

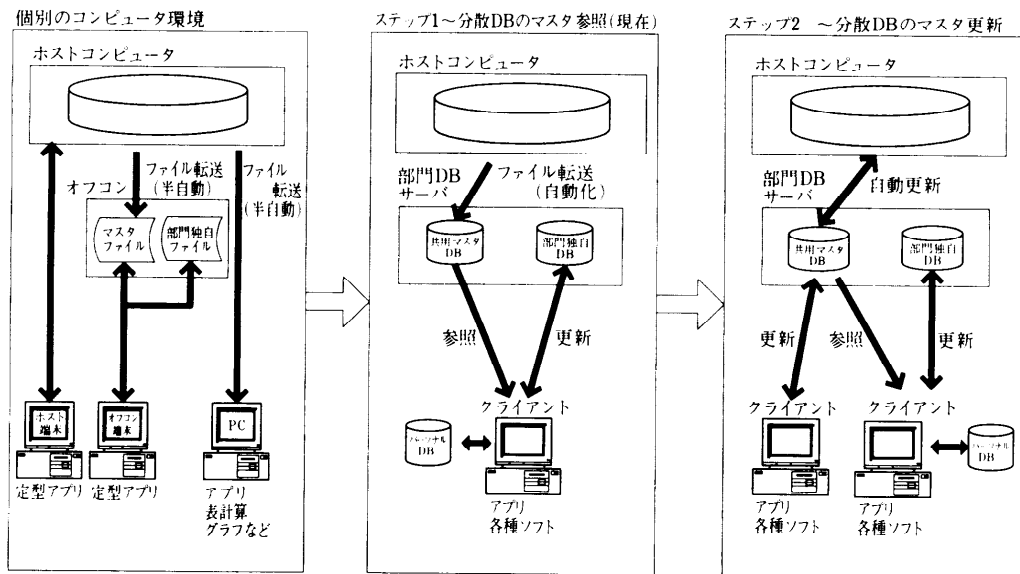


図-1 分散DB構築のステップ

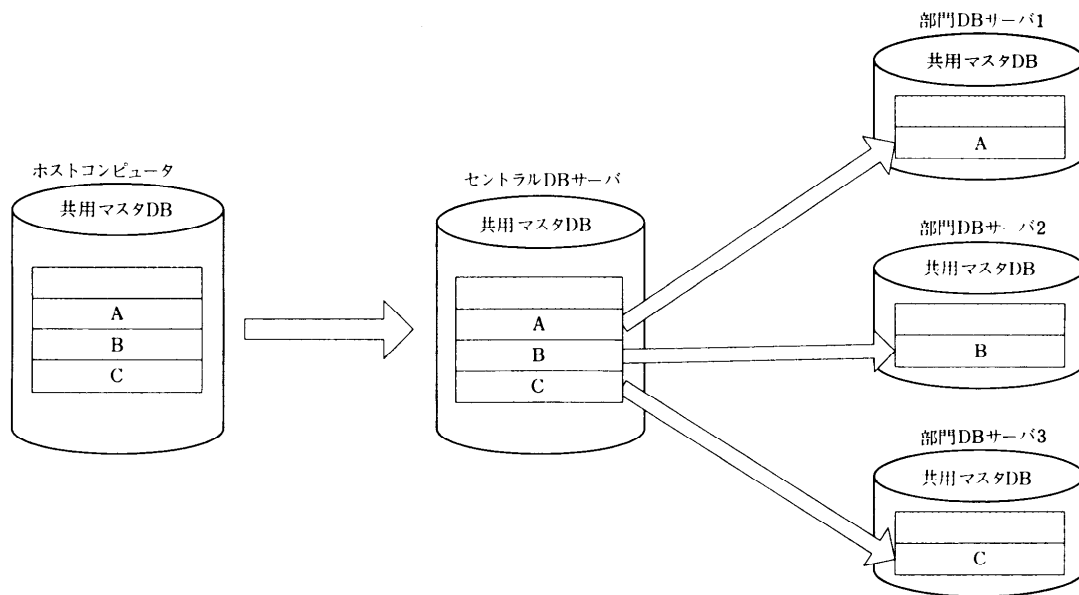


図-2 共用マスターDBの構成概念～ステップ1

3.3 DB配置の考え方

従来の開発では、データベースは個別システムごとに設計されており類似のものが複数存在する。C/S環境のアプリ開発の初期段階では、データベース全体の分類を考え共有データをマスター

として定義し、それを一元管理することによりデータの共用とDBの統合管理ができる体制とした。

DBは以下のように分類している。

- (1) 共用マスターDB

個々のアプリから参照されるDBであり、管理

単位からさらに以下のように分類される。

- ① 全社 DB …全社単位で管理されているデータ群で、顧客マスタやコードマスタなどがこれに当たる。
- ② 部門 DB …部門単位で管理されているデータ群で支払先マスタや従業員マスタなどがこれに当たる。

(2) アプリ個別 DB

個々のアプリのデータである。

ステップ1ではこの共用マスタ DB は参照利用であり、以下のルールとしている。

- ① レイアウト管理は共用マスタを管理する情報システム部で行い、個別のアプリの要求でのフィールド等の変更はしない。
- ② 情報システム部で登録・更新用のアプリを作成し提供する。
- ③ 全社 DB との整合性についても、この提供されたアプリで保証される。

この共用マスタ DB は、図-2「共用マスタ DB の構成概念」のように配置した。図の斜線の部分は全社 DB であり、概念的にはどの DB サーバにもまったく同一の内容が配置されている。

一方、部門 DB は部門 DB サーバの A、B と C である。これらは完全に同一項目からなるが、データは部門ごとに異なるものである。またセントラル DB サーバは、各部門 DB サーバの A、B、C を合わせた全社分を配置している。

4. 分散 DB 運用システムの概要

4.1 運用のための処理概要

現在のステップ1では、「ホスト」で管理されている共用マスタ DB を「DB サーバ」へ配付することと、部門に蓄積されている DB サーバと全社分のデータが蓄積されている DB サーバとの整合性をとるための分散 DB 運用システムが必要である。

[1] ホストから DB サーバへのデータ配付

共用マスタ DB のホストから部門 DB サーバへの配付は、夜間の自動処理により行っている。図-3「ホストから DB サーバへの共用マスタ DB の配付」に全体の処理の流れを示した。

以下に、処理プロセスについて示す。

- (1) DB 更新管理サーバ (C/S 側) から、

ホストの「更新データ作成処理」を起動

(2) 更新データ作成処理

- ① 更新データ作成
 - ② ホスト体系から PC 体系へデータ変換
 - ③ データ圧縮
 - ④ RJE 送信キューへファイルの出力
- (3) ホストから RJE サーバへファイル転送
 - (4) DB サーバでの更新処理
 - (5) 更新処理ログを DB 更新管理サーバへ記録

この処理結果は、DB 更新管理サーバで集中管理している。なお、DB サーバへの更新は以下の2タイプを用意している。

(1) 置換更新

- ① DB サーバのデータをすべて置き換える処理
- ② 大量データが更新されるタイミングで使用
- ③ データの物理順を整理

(2) 差分更新

- ① 一定期間内の更新分データを更新
- ② 置換更新と次の置換更新の間に行う

この処理タイプは、ホストのマスタ更新サイクルに合わせ使い分けられる。たとえば更新が集中する期間は置換処理、少量のデータ更新の期間は差分更新を行うよう運用している。

[2] DB サーバ間のデータの整合性

各部門 DB サーバに蓄積されたデータを全社分として集計するためにセントラル DB サーバに対して更新したり、全社管理業務により更新されたデータを各部門 DB サーバに対して更新する必要がある。そこで、同一スキーマのテーブル間の整合性をとるためのツールとして DDM (Deferred Data Maintenance) システムを開発した。

4.2 運用ツールの機能

分散 DB 運用システムはいくつかの運用ツールを組み合わせることにより稼動している。

[1] RJE (Remote Job Entry system)

ホストと C/S 間でデータ転送する汎用ツールとして開発したもので、双方向にファイル転送を自動的に行うシステムである。さらに、ファイル受信後に「ファイルの到着をトリガーとして定義されているプロセスを起動し、データ処理を行う」という機能を持たせた。この機能はホストと C/S の両サイドにある。図-4「分散 DB の運用のためのツール (その1 - RJE)」に概要を示し

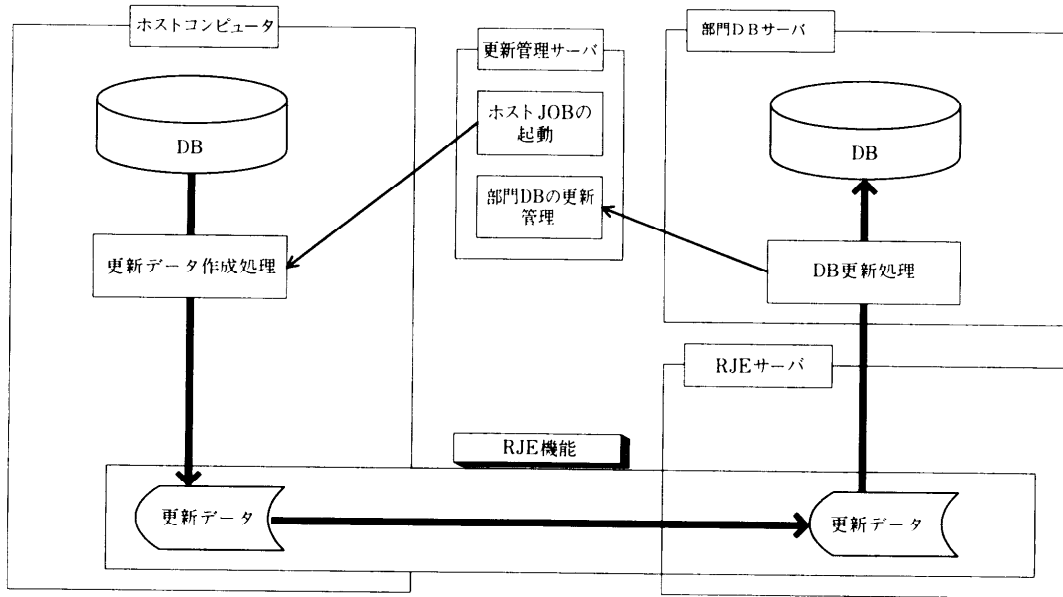


図-3 ホストからDBサーバへの共用マスタDBの配布

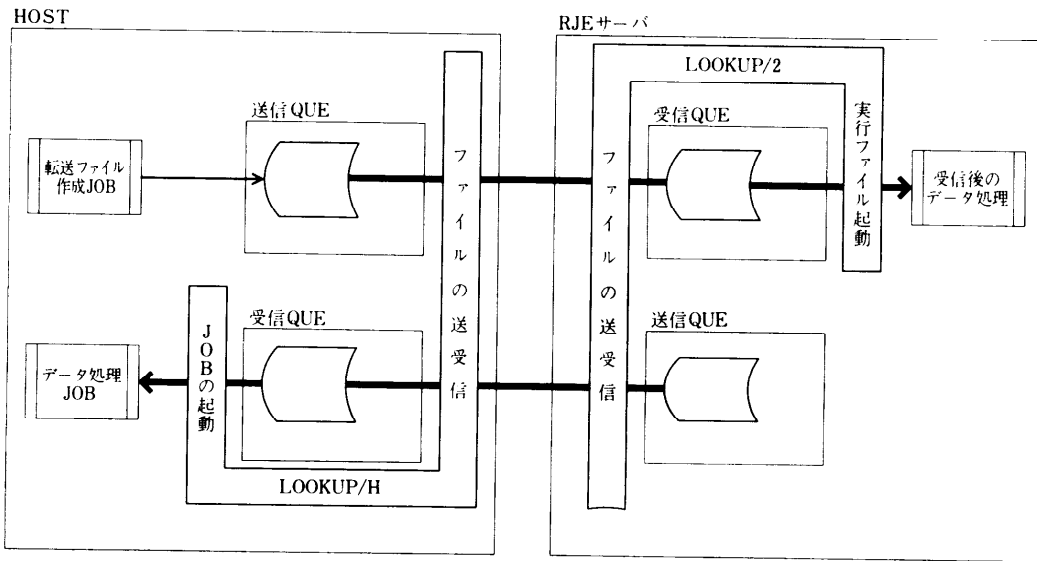


図-4 分散DBの運用のためのツールその1 - RJE (Remote Job Entry system)

た。以下にRJEの構成部品を説明する。

(1) LookUp / 2

OS / 2上でファイル到着を監視するモジュールである。監視PATHにファイルが到着すると処理モジュールを子プロセスとして起動する。し

たがって、DOSクライアントからでもOS / 2サーバに対し定義されているファイルを書き込むことでRPCを実現している。

(2) LookUp / H

OS / 2のLookUp / 2と同等の機能を持った

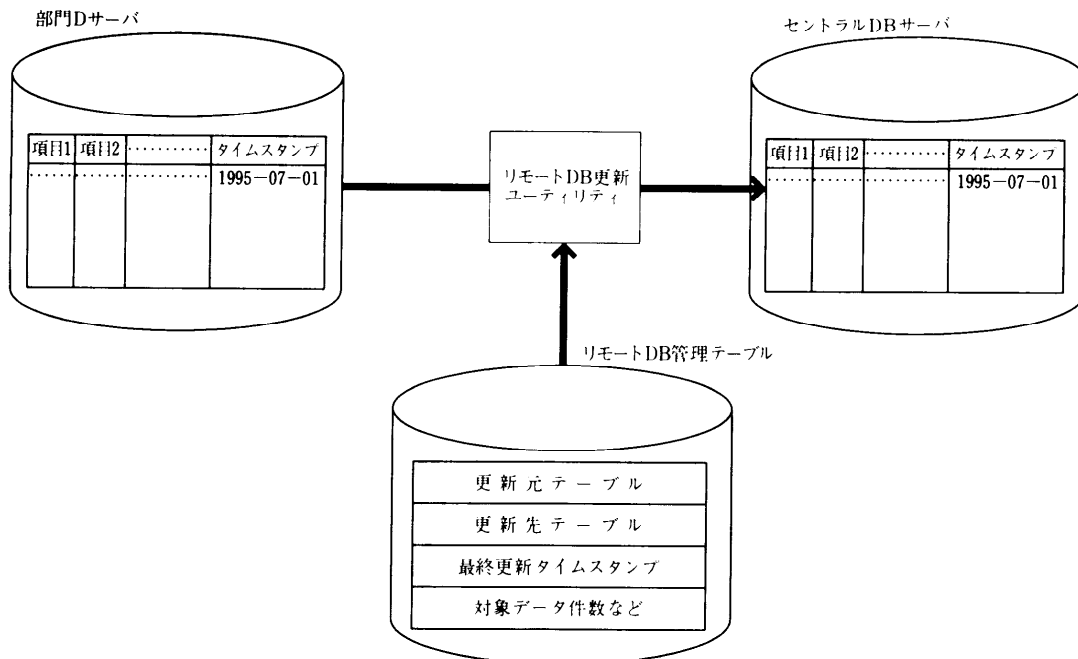


図-5 分散DBの運用のためのツールその2～DDM (Deferred Data Maintenance)

ホスト側のアプリである。

(3) ファイル送受信プログラム

ホストとサーバ間でのファイル送受信プログラムでLookUpから起動される。

[2] DDM (Deferred Data Maintenance)

分散リレーショナル・データベース体系 (DRDA) を使い異なるDBサーバ間のデータの整合性を維持するためのミラーリング・ツールとして開発したシステムである。図-5「分散DBの運用のためのツール (その2～DDM)」に概要を示した。DDMは、更新管理をするための「リモート管理テーブル」を持ち、DDMは前回更新タイムスタンプをもとに今回の更新処理を行い、結果を管理テーブルへ記録する。

[3] ONTIME (時間起動管理プログラム)

DB更新用ファイル作成ジョブやDDMを起動するためには、時間を監視し処理を起動するためのツールが必要である。そこで、OS/2上で動作する時間起動管理プログラムを開発した。

[4] データ圧縮解凍ツール

分散DBの運用にともない、ホストとPC、PCとPC間でファイルを転送する量が増加するので

ネットワーク・トラフィックを抑えるためとセキュリティの点からこのツールを用意した。

また、ホストとPCでは、データのコード体系が異なるので、ホスト上で「ホスト体系とPC体系」の双方向へのコード変換ユーティリティを開発した。ホストが絡んだファイル転送には、このユーティリティと組み合わせて使用している。

5. 開発上の苦労点と解決

(1) C/Sアプリの開発

アプリからのアクセスは自部門のDBサーバとしているが、アプリとしては接続先を柔軟に変更できるようにしなくてはならない。これは、開発用から運用用のDBサーバへの切替え、同一アプリの複数部門への展開や、運用用サーバの増設などでアプリがアクセスするサーバを容易に切り替える機能が必要となるからである。そこで、クライアントのアプリが容易にクライアント側のDBインタフェース情報を更新できるようにDLLを用意しこの機能を共通化した。

(2) 分散DB更新処理の自動化

分散DB間のデータ更新処理の開始は、

- ① 更新データが発生したらすぐに処理
- ② スケジュールにより順番に処理

できるようにする必要がある。そこで、設定時間がくると処理をキックする ONTIME と監視パスへのファイルの書込みをトリガーとし処理を起動する LOOKUP を開発した。

(3) DB 間の整合性の取り方

データの整合性をとるための処理は、更新時のデータ（項目単位）で行う場合、トランザクションデータを保持し、他の DB（テーブル）に対してそれを実行することになる。しかし、テーブルに対しての更新命令すべてのトランザクションを DB サーバ上で保持するのは非常に難しい。

そこで、DB サーバ間の整合性をとる DDM では、データの整合性はテーブル単位で行うこととし、相手のテーブル上のデータ有無はユニークキー・カラムでチェックしている。アプリでは DDM の処理対象となるデータを判断するため、アプリの更新時にタイムスタンプをセットする。

また、ホストから DB サーバへの差分（ある期間の変更分）更新では、ユニークキーを使った SQL の DELETE 文とレコード単位のインポート処理を使用するようにした。

(4) 更新処理結果の集中管理

夜間に分散 DB の同期処理を行っているが、処理結果を 1 台ずつ確認するのは困難である。更新記録データを管理サーバの DB へ集め異常結果を検知するようにしている。

6. システム運用後の効果

現在の分散 DB は、構築ステップ 1 の「分散 DB マスタの参照」という状況ではあるが、社内で共用度の高い顧客、取引先、会計口座や人事マスタなどが自動的に整合性がとられているので、個別のアプリは開発においてアプリで蓄積する DB だけを作成すればよいので開発効率がよい。

また、業務用に開発されたアプリは統一された環境で稼動するし操作の原則も市販ソフトと同じようにできるので、ユーザがホスト系、オフコン系などということ意識することなく使用することができる。

また、自部門の業務に必要なデータが DB サーバに蓄積されていくのでユーザ自身が業務で蓄積されたデータを簡易ツールなど活用が図れるよう

になり効果を上げている。

7. 今後の課題

今後は、マスタを更新する業務を C/S 環境へ移行して、データ更新が行われたらすぐに他の業務から参照できる状態を構築していきたい。また、現在は個別 DB の中にあるデータでも、新規に開発されるアプリから参照されるデータとなるかもしれない。そこで、各々のアプリ用の個別 DB に関してもデータの 1 次更新、2 次更新先などの関係を常に明確にしておける管理が必要である。

したがって、分散 DB のデータ資源を管理するポイントは「DB の配置（オリジナルと複数のレプリカ）」「整合性のタイミング」と考えられ、これらに関する管理体制を充実していきたい。

現在の運用ツールで、ホストと C/S の DB の整合性をとる仕組みはファイル転送を使用しているが、このをトランザクション・データにより更新ができるようにしていきたい。これは DBMS の提供する機能に依存する部分が多いが、常に最新の技術動向を注意しながら進めたい。

8. おわりに

C/S でのアプリや各種のソフトからのデータ活用を効果的に行えるようにするために分散 DB を構築、運用してきた。部門ユーザは、利用頻度が多いマスタ類が提供され、データも自動更新されていることで大きな効果を上げている。さらに利用できる情報源の拡大を目指し、その効果を高めて行きたい。

(平成 8 年 1 月 17 日受付)



小林 正美

1960 年生。1983 年東京理科大学理工学部経営工学科卒業。同年清水建設（株）入社。入社後、情報部門へ配属。メインフレームにおける工事実績や技術情報などの検索システムやパソコン版の海外拠点向けの会計システムなどの開発業務を行う。その後、メインフレームの DBMS 運用業務を経て、現在、情報システム本部情報システム部に所属し、C/S 環境における DB の運用業務に携わる。