

6E-4

# 異種データベースシステム接続環境での OSI-RDA実装方式について

山本 裕 石川 博道

(株)日立製作所

## 1. はじめに

ダウンサイジング化やデータ連携機能仕様のオープン化が進み、ユーザ環境では従来のメインフレーム-端末接続のデータ連携から、WS、PC主導型コンピューティング環境での分散DB処理への移行が増加している。

分散DB基盤の汎用化、多様化、通信プロトコル制御及び異種DB接続実装技術の向上が求められる中で、日立では分散DB接続のDB種別の拡張と分散DB機能の汎用化を目的とし、OSI-RDAを適用してメインフレーム、WS及びPCを接続した環境での異種DB接続を実現する。

本論では、日立分散DB機能におけるOSI-RDAを用いた異種DB接続方式について述べる。

## 2. 日立分散DB機能の概要

日立の分散DB機能は、日立メインフレーム、WS及びPCを接続したネットワーク環境で実現する。図1に日立分散DB機能構成を示す。本機能では、PC、WS及びメインフレーム上のAPから同一のインタフェースでサーバのDBアクセスが可能となる。異種DBとの接続性、分散DBプロトコル機能の汎用性を持つ本分散DB機能の特長を示す。

(1) OSI-RDAプロトコルの実装

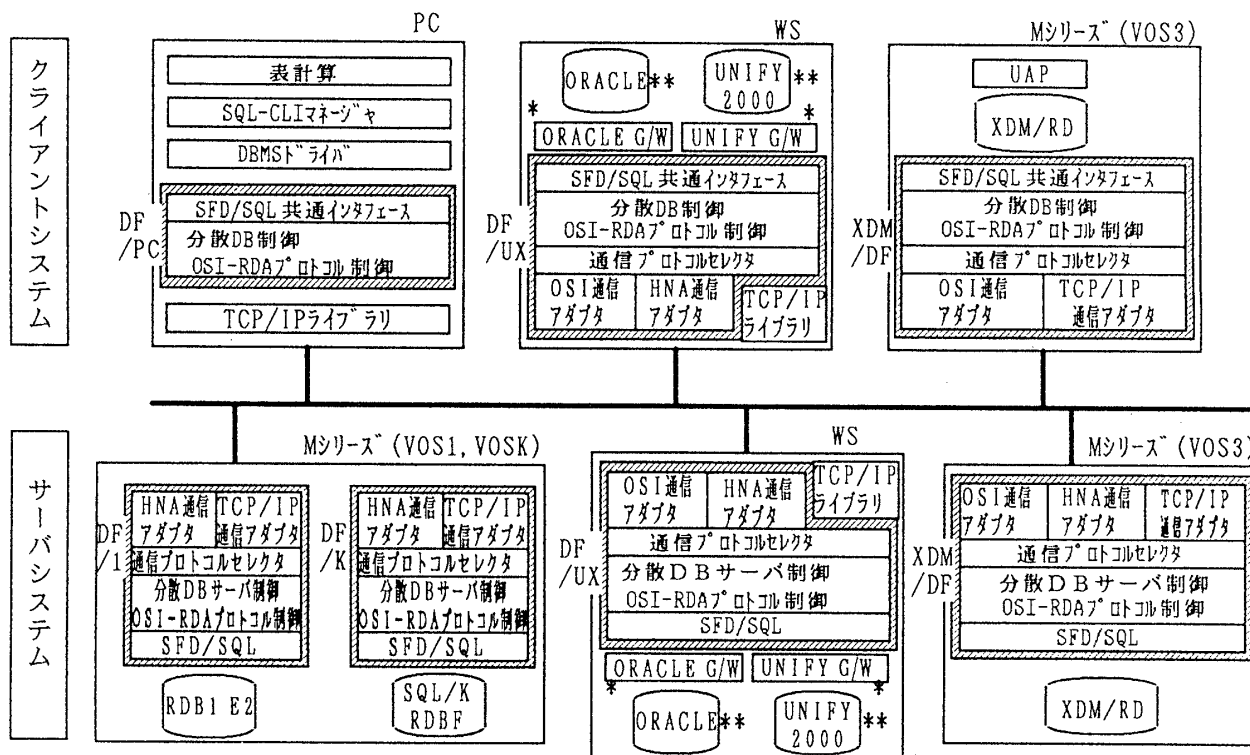
異種DB接続プロトコルであるOSI-RDA標準プロトコルを実装。

(2) マルチプロトコルの共存

OSI、TCP/IP及びHNAの各通信プロトコル上にOSI-RDAを実装し、選択及び共存を可能にした。

(3) 複数応用コンテキストの共存

接続性を確保するため、通信プロトコルに



\* ORACLE, UNIFY2000各々のG/Wは、日立分散DB機能共通 (SFD/SQL共通)インタフェースと各DBMS固有のインタフェース間のゲートウェイ機能を持つ

\*\*ORACLEは、米国Oracle Corporationの登録商標です。

\*\*UNIFY2000は、米国Unify Corp.の商品名称です。

図1 日立分散DB機能構成

関して、複数の実装レベルを複数の応用コンテキストとして設定し、共存させた。

(4) SQL環境の設定

各DBMSが適用するSQL仕様には差異があり、接続するDB同志が認識するSQL環境レベルを設定し、接続時に相互折衝する方式を採った。

(5) プロトコル制御機能の汎用化

OSI-RDAにマッピングしたインタフェースを共通化してDBMSに見せている。

3. 異種DB接続方式

3.1 DB接続の意義

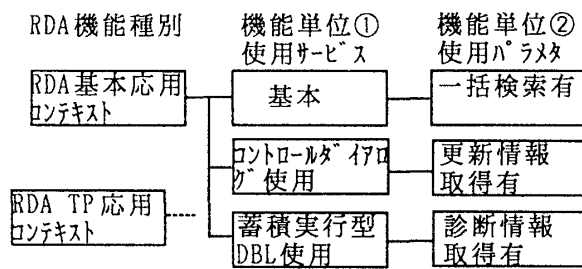
図1のような環境で分散DB処理を行う場合、接続相手との間で相互にプロトコル実装レベルや適用SQLレベルの折衝が必要となる。実際に折衝を行う項目は、

- ・通信プロトコル
- ・応用コンテキスト(機能単位, 実装レベル)
- ・SQL環境(データフォーマット, 文字コード仕様)

がある。通信プロトコルに関しては、相互に同じである必要があるが、応用コンテキスト及びSQL環境に関しては、接続する一方が他方の機能仕様のサブセットである場合には、低い方のレベルの機能仕様で接続可能となる。接続仕様をレベル分けすることによって、機能を縮小した範囲での接続を許容することになり接続性が拡張する。

3.2 具体的な接続方式

接続性を確保するために、応用コンテキスト及びSQL環境のクラス分けを行い、クラス間関係を定義する。応用コンテキストを例にとり接続方式を述べる。図2に示す様に各々の応用コンテキストには、RDA機能種別、使用サービス種類、パラメタ種類が異なるレベルが存在する。各レベルには親子関係があり、この関係に従って図2の様にクラスを作成し、関係を定義する。



【クラス分類】 RDA基本応用コンテキスト

- クラス1: 基本 R-Initialize, R-Open, R-Close, R-Begindialogue, R-ExecuteDBL, R-Commit, R-Rollback, R-Terminate R-Synchronize サービスを適用
- クラス2: クラス1 + コントロール (R-Cancel, R-Status) タプル使用
- クラス3: クラス1 + 蓄積実行 (R-DefinedBL, 型DBL, R-DropDBL, R-InvokeDBL)
- クラス4: クラス2 + クラス3

図2. 応用コンテキストのレベル分けの方式例

応用コンテキストに関しては、通信路確立時にクラスを示すオブジェクト識別子(コード)を用いて相互に折衝する。同じクラスの場合は接続可能となる。異なったクラスの場合、図2の例では、排他関係にあるクラス2とクラス3では接続不可とし、クラス1とクラス2の様に包含関係にあるクラス同志は、サブセットとなるクラスのレベル(クラス1)で接続する。この様にクラス間関係を相互に定義し、定義のルールに従って接続する。

3.3 折衝方式

前記のクラス定義を使用した折衝方式例を以下に示す。

- ① 予めクライアント及びサーバにクラス定義を保持する。(同一の定義)
- ② クライアントより自クラスで接続要求。
- ③ サーバではクライアントの接続要求に対して相手のクラスによりクラス定義を元に接続判定を行う。相互のクラスが異なる場合、接続不可であり再接続要求を促すか、接続可能なクラスで接続を実行完了した旨をクライアントに回答する。
- ④ クライアントは自接続要求に対するサーバの回答に従った振舞をする。

表1に相互接続時に折衝を行う項目と折衝時期を示す。各々の折衝項目毎に前記の様な相互折衝を行う。

表1. 折衝項目と折衝時期

折衝項目	折衝項目詳細(親子関係)	折衝時期
	親 ← → 子	
応用コンテキスト	RDA機能種別 ← 使用サービス種類 → 使用パラメタ種類	アソシエーション確立時
SQL環境	SQL環境種類 ← 使用SQL種類 → 使用データフォーマット	DBオープン時

4. 終わりに

本稿では、メインフレーム、WS及びPCを含めたネットワーク環境での分散DB機能における異種DB接続方式について述べた。

本方式により、接続性の拡張を図ることができるが、今後、前述のようなクラス定義を始め分散DB機能が使用するネットワーク環境の定義等の運用管理技術の向上に努めると共に、さらに分散DB機能の拡張を図るべく標準仕様の実装に対応していく。

参考文献

- [1] ISO/IEC 9579-2:1993 Information Technology - Open Systems Interconnection Remote Database Access Part 2:SQL specialization