

# 部門内処理向け分散RDB管理システム

## 3H-3

岩井 徳幸\* 加納 直紀\*\* 青木 久延\*\*\* 西脇 孝一\*\* 加藤 正道\*  
 \*(株)日立製作所システム開発研究所 \*\* (株)日立製作所ソフトウェア工場  
 \*\*\* 日立マイクロコンピュータエンジニアリング(株)

### 1. はじめに

最近のLANとワークステーションの普及により、LANに接続したワークステーション間で、データの共有を実現したいというユーザの要求が高まってきた。我々は、この要求を満たす分散RDB管理システムの開発を行った。本稿では、分散RDBの開発方針を述べた後、システムの制御方式について報告する。

### 2. 分散RDBの開発方針

スタンドアロンで提供されるRDBの機能をベースに、その機能の外側に分散化機能を付加し、表1に示すような比較的小規模で、基幹業務との関連の小さい環境へ適用できるシステムの実現方式を検討した。

このためには、少なくとも次の要求を満たす必要があると考えた。

- (1) ロケーショントランスペアレンシの実現。
- (2) 1コマンドのアクセス対象は、1データベースとする。
- (3) 簡便な運用操作機能を提供する。

エンドユーザが自らシステムを運用する環境では分散RDBの基本機能と共にシステムの運用を簡便にすることが重要であると考えられる。今回の報告では、(1)、(2)を中心に述べ、(3)は別途、報告する予定である。

表1 適用対象

項番	項目	内容
1	同時アクセス	数十～数十台のクライアントが、1台のサーバをアクセスする
2	ホストとの関連	ホストDBとの関連は密とせず、アップ/ダウンロードにて対応
3	扱うデータ量	テーブル当りのロー(レコード)件数が、数千～数万
4	サーバのアクセス形態	トラヒックは大でなく、検索が中心
5	システムの運用	エンドユーザ自身(専門家ではない)

### 3. システムの構成方式(図1)

ドメインと呼ぶ論理的な管理単位を設け、1つ以上のドメインからシステムを構成するものとする。これにより、ドメインを追加することでシステム構成を拡大でき、管理はドメイン単位で閉じたものとする。その影響が全体に波及しないようにでき、運用が容易になる。

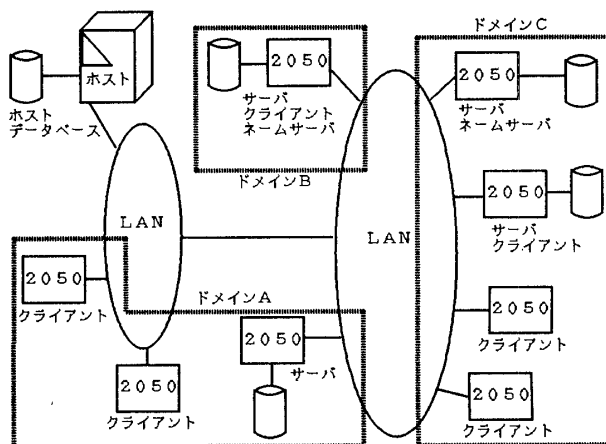


図1 分散RDBシステム構成例

### 4. 制御方式

#### (1) サーバ/クライアントモデル

分散DBアクセスの仕組みは図2に示すように、アクセス要求側と処理要求側からなる、サーバクライアントモデルで実現する。これにより、

- ①クライアント専用ステーション  
(リモートアクセス部のみ存在する)
- ②サーバ専用ステーション  
(ローカル処理が存在しない)
- ③クライアント・サーバ共存ステーション  
(両方の機能を持つもの)

を構築することが可能となり、分散処理システムの多様な形態に対応できる。

Distributed RDB Management System for Work Group Computing

Noriyuki IWAI\*, Naoki KANO\*\*, Hisanobu AOKI\*\*\*, Koichi NISHIWAKI\*\*, Masamichi KATO\*

\*System Development Lab., Hitachi, \*\*Software Works, Hitachi,

\*\*\*Hitachi Micro computer Engineering

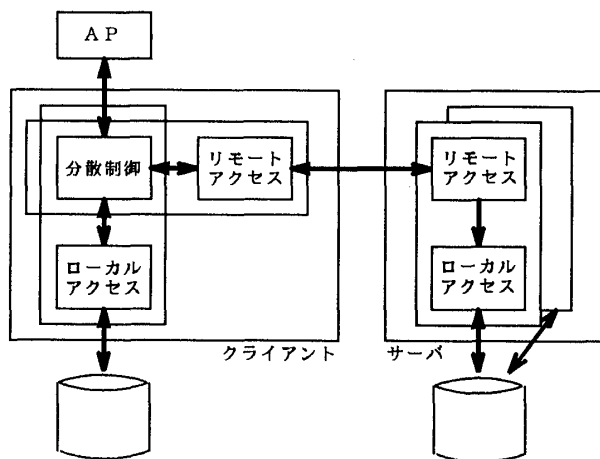


図2 サーバ/クライアントモデル

## (2) 分散データベースの実現形態

分散データベースの実現形態として、サーバの持つデータベースを構成するファイルを、クライアントがシェアすることでも実現可能である。しかし、この方式は、ファイルI/Oのレベルで通信が多発し、効率を低下させる危険性がある。そのため、SQLを用いた高い機能レベルでサーバとクライアントが通信を行う方式を採用した。

## (3) ロケーショントランスペアレンシ

### (a) テーブル名の拡張

テーブル名指定の構文は、以下に示すようにデータベース名による修飾を許す形式とし、これにより、データの存在するノードを意識せず、論理的な名称で対象テーブルをアクセス可能とした。

テーブル名指定の構文

[データベース名.][認可識別子.]テーブル名

### (b) データベースアドレス情報

クライアント側に、データベース名とデータベースの存在するノードのワークステーションアドレスの対応表を持つ。

以上述べた方法により、ロケーショントランスペアレンシを実現した。さらに、運用のためのユーザ負担を軽減する目的で、ネームサーバによるアドレス情報の一元管理を行う方式にする予定である。

## (4) マッピングによるユーザ管理方式

我々は、各ワークステーションでのユーザ管理は必ずしも一致してはいないという前提に立ち、先に述べた、システムの最小の管理単位であるドメイン

内でユニークとなるグローバルユーザIDを導入した。このグローバルユーザIDと、各ワークステーションで管理されるローカルなユーザIDをマッピングする事によりユーザ管理を実現した。また、グローバルユーザIDのユニーク性は、運用者の責任において満たすものとし、将来的には、これをネームサーバの管理により行う予定である。

## (5) アクセスインタフェースとプロトコル

APインタフェースはJIS準拠のSQLを使用している。クライアント/サーバ間のインタフェースは独自のプロトコルにより、SQLコマンドストリームをサーバへ送り、その結果をクライアントへ返す方式とした。

また、検索で得られる複数のレコードを一括して送ることにより、サーバ/クライアント間の通信回数を削減する方式を提供している。

## (6) 効率の良いレコード排他制御方式

マルチユーザで利用する際、アクセス競合による性能劣化を防止するため、より小さな単位の排他制御が重要となる。そこで我々は、ロー（レコード）レベルのロック機能を提供することとした。ローレベルのロック制御により生じる管理オーバーヘッドを低減させるため、ロック情報をメモリ上に常駐させ、容量不足の時はファイルに格納する階層管理方式を採用した。

## 5. おわりに

本稿では、主として、部門内分散処理システムに用いる水平分散RDBの基本制御方式のを中心に述べた。今後、エンドユーザの運用操作性の向上を図るネームサーバ技術や、性能向上、運用の容易化などの改善を図っていく予定である。

## 参考文献

- [1] “ネットワーク環境に対応、クライアント/サーバ型に進化するデータベース”，  
日経コンピュータ，1989.9.25，pp.103-113
- [2] “データベース言語SQL”，  
JIS，X3005，1987