データベース・システム 83－4
（１９９１．５．２４）

垂直分散データベースにおける問合せ最適化処理の実現方法

高野 智　鶴岡 邦敏

日本電気㈱Ｃ＆Ｃシステム研究所

ローカルデータベースのデータ間に，階層的な従属関係が成り立つ分散データベースを「垂直分散データベース」と定義し，研究開発を行っている．本稿では，現在開発中の垂直分散データベース管理システムで採用する管理方法と問合せの最適化方法について論じる．管理方法では，主にデータベース間に存在する重複データの管理について説明する．問合せ最適化では，準結合(semijoin)処理の他に，問合せ発生サイトに存在する重複データを利用した転送データ量の削減方法について説明する．

ON QUERY OPTIMIZATION IMPLEMENTATION FOR VERTICALLY DISTRIBUTED DATABASES

Satoshi Takano　　Kunitoshi Tsuruoka

C&C Systems Research Laboratories, NEC Corporation

4-1-1 Miyazaki, Miyamae-Ku, Kawasaki, Kanagawa 216, Japan

We define "vertically distributed databases" as the distributed databases with dependence relationship between local databases. In this paper, the data management and the query optimization for vertically distributed databases are discussed. The management of replicated data distributed in several local databases is mainly discussed in the data management. In the query optimization, we use the replicated data in the local database in addition to the semijoin method in order to reduce the data transmission cost. The schemes proporsed here are being implemented for the prototype system.

1．はじめに

情報化社会の進展と共に，集中的なデータ管理にかわり，企業内の各部門ごとに必要なデータを蓄積・管理し，他の部門のデータはネットワークを介してアクセスする作業形態が増えてきた．このような市場の状況に加えRDA（リモートデータベースアクセス）の国際標準化も進み，データベース環境が集中型から分散型へと移行しつつある．分散データベースに関する研究および開発は以前から行われていたが，ほとんどの場合，データの管理は所在データベースの管理システムに任されており，個々のローカルデータベースは対等の関係であり，相互に通信ができるというもの以上の関係は何も定義していなかった．また，分散データベース内に存在するリモートデータベース間の重複データの問題は，極力取り扱うことを避け，重複データがないことを仮定して議論が進められることが多く見受けられた．

しかし，現実の社会あるいは企業の組織管理と情報の利用形態とを考えると，個々の部門にはある種の上下関係が存在し，それにともなって情報利用の局所参照性やリモートデータベース間の重複データの存在が考えられる．したがって，分散データベースを実際の業務へ適用するには，従来の分散データベースの管理に加え，各部門の上下関係あるいはそれらが連なった階層関係から生じるデータの利用形態に適した管理をする必要がある．我々は，従来の分散データベースの管理を「水平分散データベース」の管理とし，それに対して各ローカルデータベース間の階層関係を考慮に入れた管理を「垂直分散データベース」の管理として位置づけ，研究開発を行っている．

本稿では，最初に垂直分散データベースの基本モデルを説明し，次に現在開発中の試作システムについてシステムの構成や管理情報などを紹介する．そして最後に，分散データベースには重要な機能となるリモートデータベースの表に対する問合せの最適化について，本試作システムで採用している手法を紹介し，その実現方法を説明する．

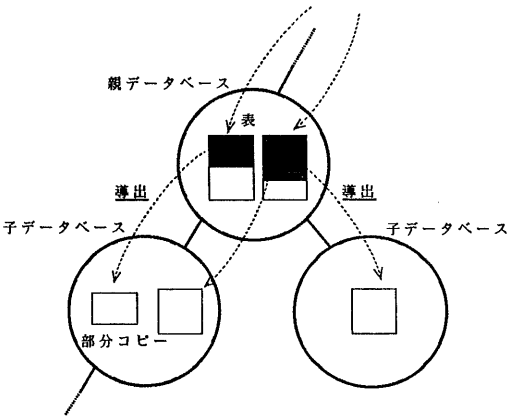

図　1　データ分散の基本モデル

2．垂直分散データベースとは

「垂直分散データベース」とは，相互に接続されたローカルデータベース間に何らかの上下関係が存在するものとして定義される．何を基準として上下関係を定義するかはさまざまな解釈が可能であるが，我々はデータの「従属関係」をデータベースの上下関係と定義している[1]．従属関係をもつものの例として，上位部門が下位部門のデータを集約して保持したり，逆に上位部門で生成または集約されたデータの部分コピーを下位部門で利用するという形態が考えられる．

図1に垂直分散データベースのデータ分散の一つのパターンを示す．リモートデータベースの表の部分コピーを自データベースに作成することを，「導出」と呼ぶ．導出により重複データが作られた場合，導出元と導出先の表には「導出関係がある」という．導出表から，さらに導出して別のリモートデータベースに表を作成することも可能である．次々に導出が繰り返されると，データの導出関係が全体として階層をなす．このように，階層化した重複データの系列を従属関係と呼ぶ．本稿では，ローカルデータベース間のデータに従属関係が存在する分散データベースを，「垂直分散データベース」と定義する．

3．プロセス構成

各ローカルシステム内には，ローカルDBMSと垂直分散DBMS（垂直システム）のサーバプロセスが常駐している．利用者アプリケーション（ＡＰ）が垂直システ

ム起動関数を呼び出すと，垂直システムのクライアントが主記憶にロードされる．そして，プロセス間通信を用いて利用者ＡＰからの要求を受け取り，要求に応じた処理を実行した後利用者ＡＰに処理の終了通知を返却する．

リモートアクセスが生じた場合は，所在情報を参照し，当該サーバへ通信要求を出す．リモートデータベースのサーバは，通信路を確立した後，パラメータを受け取り，要求された処理に対応した関数を呼び出す．

このようにして，垂直システムは利用者ＡＰから終了要求が出されるまで処理を繰り返す．利用者ＡＰから終了要求が出されると，垂直システムはデータのセーブや記憶領域の解放などの後処理を実行して，プロセスを終了させる．

本システムでは，一つのプロセスの起動から終了までの間に得られた管理情報や問合せ結果の一時表などを保持し，後の問合せに利用して，冗長な処理を極力防ぐようにしている．

4．管理情報

4.1 システム構成表

垂直分散データベースを構成するローカルデータベースシステムには，データの重複従属関係に応じて，ある種の上下関係が存在していると仮定する．そのローカルデータベースの上下関係が連なることにより，システム全体が階層化された木構造をなしていると仮定できる．本システムでは，このシステム全体の木構造を，垂直分散データベースの管理情報として関係データベースのフラットな表として登録するために，システム構成表という管理表を定義している．システム構成表の各タプルは，以下のような属性からなる．

データベース名	レベル	親データベース

データベース名：ローカルデータベースのグローバルデータベース名．

レベル：ローカルデータベースが位置する，垂直分散データベースの木のトップからの段数．

親データベース：ローカルデータベースの直上に位置するデータベースのグローバルデータベース名．

システム構成表には，各ローカルデータベースが垂直システム全体の最上位データベースから何番目に位置し，そして直上のデータベースの名前は何かという情報が示される．これらの情報は，主に垂直システム内の重複データの探索に利用される（5.3で述べる）．

4.2 所在管理辞書

分散データベースでは，データの移動や格納サイトの名前の変更のたびごとに利用者ＡＰの修正が生じると非常に不都合である．したがって，データの格納場所やリモートデータベースのシステムの変更を，利用者に意識させないこと（位置の透過性）が必要となる．位置の透過性を実現するために，分散データベース管理システムでは，システムが参照する管理情報としてデータの所在情報を持たせる必要がある．

開発中の試作システムでも，データの所在に関する情報を管理するための所在管理辞書を持つ．所在管理情報は，垂直システム起動時にローカルデータベースの所在管理辞書から，垂直システムを起動させた利用者が利用できるデータベースに関するものだけを取り出し，高速アクセスのためのハッシュテーブルを作成して主記憶に取り込まれる．垂直システムの各機能では，ハッシュテーブルを介して必要な表の所在情報を得る．そしてこの所在管理情報は，利用者ＡＰから終了要求が来るまで主記憶に保持される．

4.3 重複データ管理

開発中の垂直システムの特長の一つは，分散データベース内のデータの重複従属関係を管理することである．重複データに関する情報を一括して一つのデータベースで管理すると，重複データ情報の参照や新たな重複データの登録によるリモートアクセスが頻繁に生じる恐れがあり，一つのデータベースへのアクセスが集中し，リモートアクセスの応答時間がさらに悪化することが予想される．一方，すべての重複データ情報をすべてのデータベースに格納する方法をとると，情報を得るための処理は高速化できるが，重複データ情報の更新がシステム全体に及び，各々のシステムの利用者へのサービスに影響を与えてしまう．また，各ローカルデータベース内に，そのデータベースからほとんどアクセスしないデータに関する重複データの情報

も格納されるため，格納領域も無駄である．

本稿で定義した垂直分散データベースの環境下では，システム構成上比較的近くに位置するデータベースへのアクセスが多い（局所参照性がある）と仮定できる．つまり，リモートアクセスの内で最も多いのは隣接データベースへのアクセスであり，そこから離れるにしたがって徐々にアクセス頻度が低下すると予想される．したがって，本システムでは隣接データベース間で相互に導出データの情報を管理することによって，全体の重複データを管理する方法をとる．各データベース内に導出元管理表（隣接データベースから抽出した表の名前を登録した表）と導出先管理表（自データベースから隣接データベースへ抽出された表の名前を登録した表）を作成し，それぞれ自データベース内の表と隣接データベース内の表との重複関係を管理する．局所参照性により，リモートアクセスの多くは隣接データベースへのアクセスであるので，リモートアクセスが発生しても，多くの場合自データベースに存在する管理表のローカルな検索で，リモート表とローカル表の重複データ情報が取り出せる．

導出元管理表と導出先管理表の構成はほぼ同じであるので，導出元管理表についてのみ説明し，その後に重複データの生成方法について説明する．

• 導出元管理表

導出元管理表は，リモートデータベースから表の一部を導出し自データベースに格納した重複データに対して，そのデータの導出元を登録した表である．以下に導出元管理表に含まれる属性を示す．

表　名	導出元表名	データベース名

主キー属性	導出の型	導出規則

表名：自データベースにある重複データの表の名前．
導出元表名：導出元の表の名前．
データベース名：導出元表が存在するデータベース名．
主キー属性：導出表のタプルを一意に識別するための属性．
導出の型：導出の型を四つに分けて登録する．
　分散型：一つの表の一部を取り出し，一つの表を作成．
　集約型：複数の表からデータを取り出し，unionをとって一つの表を作成．
　結合型：複数の表からデータを取り出し，joinをとって一つの表を作成．
　サマリ型：導出時にある操作を施し，データを加工して格納．
導出規則：導出元表からどの部分を取り出すかをSQLのselect文で登録．もし，自データベースの表を仮想表（分散ビュー）形式で作成したい場合は仮想表定義SQL文を登録する．

• 重複データの生成方法

重複データの生成（導出）は，リモートデータベースの表の必要な部分を検索SQL文（導出規則）で表わし，システムのユーティリティを用いて，導出規則をリモートデータベースに転送して検索処理を依頼し，その検索結果を受け取ってローカルデータベースに別の表として格納するという方法で行われる．導出処理を行なう前に導出元管理表に導出したいデータに関する情報を登録する．導出処理を行なうユーティリティは，導出元管理表の指定されたタプルを取り出し，導出元となるデータベースと表に関する情報を得る．そして，そのデータベースに導出規則を転送し検索処理を依頼する．次に，検索されたデータを収集し，新たな表として定義する．最後に，導出元データベースの導出先管理表に，導出された表に関する情報を登録する．

5．垂直分散データベースの問合せ処理

リモートアクセスは，一般的にローカルアクセスに比べ，コストが高く，処理にかかる時間も長い．それは，処理の途中でデータ転送の処理が入るためである．データ転送のコストとそれにかかる時間は，転送データ量にほぼ比例していると考えてよい．したがって，本システムでは，リモートデータベースへの検索要求（リモート検索）の際に，冗長なデータ転送を極力排除しリモート検索のコストの低減と処理の高速化を実現するために，問合せの最適化処理を行っている．具体的には，以下の要素を考慮に入れてリモート検索の最適化を行なう．

①分散結合（join）の最適化

結合処理では，処理前の総データ量と処理結果のデータ量が大きく異なる場合がある．分散環境で結合処理を行なう場合，最終結果として残らないデータはなるべく転送しない方が望ましい．したがって，本システムでは，分散結合の要求が生じた場合は，各表のサイズなどの管理情報をもとに，転送データ量がなるべく小さくなるような処理スケジュールを決定する．

②ローカルデータ優先検索

リモート検索の場合，指定されたリモート表と重複する部分のデータを持つ表がローカルデータベースに存在すれば，そのデータを利用することにより，ローカル検索されたデータの量だけ転送データ量を削減することができる[2]．本システムでは，リモート検索の際に，指定されたリモート表の重複データを探索し，もし重複データが自データベースに存在すれば，重複データ以外のデータだけをリモート検索するような処理スケジュールを決定する．

③処理の並列化

複数のリモートデータベースの表の検索の場合，各リモートシステムでの処理は独立に実行できる．したがって本システムでは，転送データの低減の他に，各リモートシステムごとに処理の並列化を行い，一つの問合せの応答時間を短縮するようにしている．

5.1 処理の流れ

利用者ＡＰからSQL文が転送されると，まず簡単な構文解析を行なう．もし，入力SQL文がunionを含んだ問合せであるならば，unionで連結されている各々の問合せを共有メモリに格納する．そして，unionの数だけ子プロセスを生成し，それぞれのプロセスに共有メモリに格納されている問合せSQL文を一つずつ割り当てて，各々の問合せの処理を並列に実行する．

各プロセスでは，問合せに含まれる表の所在を調べ，リモートデータベースの表があれば，問合せ最適化処理関数を呼び出す．リモートデータベースの表が含まれない場合は何もせず，プロセスを終了させる．

図２に問合せ最適化処理の大まかな処理の流れを示す．まず，SQL文で書かれた問合せを関係演算に変換し，さらに木構造（問合せ木）に変換する．そして，この

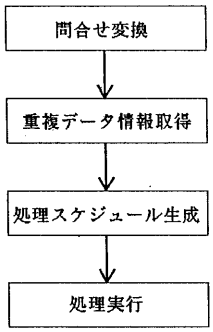

図　2　問合せ最適化処理の流れ

問合せ木を変形して，単項演算（選択，射影）をなるべく早い時期に処理できるようにする（問合せ変換処理）．次に，自データベースに存在する重複データを利用するために，重複データ情報を取得する．そして，各表のレコード数，自データベース中の重複データの有無などを考慮にいれ，処理スケジュールを決定し，そのスケジュールにしたがって処理を実行して自データベースに処理結果の一時表を作成する．最後に，自データベースに作成した一時表を検索するためのローカル検索SQL文を生成し，元の問合せSQL文を生成したSQL文に置換して，子プロセスの場合はプロセスを終了する．

親プロセスでは，全子プロセスの処理の終了を待って，共有メモリからSQL文を取り出し，それぞれをunionで連結した後，ローカルデータベース管理システムに最終的なSQL文を渡して処理を依頼する．

5.2 問合せ変換処理

SQL文は，データの操作を利用者に分かりやすい表記で表わしたものであるので，最適化を行なうのに適した形式とは言えない．本システムでは，利用者ＡＰからのSQL文を関係演算で表わし，さらに木構造（問合せ木）に変換して，その問合せ木を変形することにより最適化を行なう．本節では，本システムで採用している問合せ変換方法と問合せ木の変形方法について説明する．

①SQL文→関係演算

問合せ変換の第一段階は，SQL文を解析し関係演算を認識することである．本システムでは，選択(select)，

射影(project)，結合(join)，併合(union)の４種類の関係演算を扱う．最適化ルーチンに渡されるSQL文は，式１の例で示されるような単一のSELECT文である．これらのSQL文には併合は含まれないが，最適化の処理中に使用する場合があるのでここにあげてある．

SELECT π FROM T1, T2

WHERE A=B AND C=α; （式１）

πは属性の集合，T1，T2は表名，A，B，Cは属性名，αは定数を表わす．

一般的に，射影は検索SQL文のSELECT句，選択，結合はWHERE句に現れる．選択と結合の区別は，比較式の中に定数を含むかどうかで認識できる．問合せの対象となる表（オペランド）は，FROM句に示される．このようにして認識した関係演算子とオペランドをポーランド記法の順序に並べる．式１を関係演算に変換すると次のようになる．

PROJ.π(SEL.C=α(JOIN.A=B(T1, T2))) （式２）

②関係演算→問合せ木

SQL文から関係演算への変換が終了すると，次は問合せ木を組み立てる．本システムで取り扱う関係演算子は，単項または二項演算子であるので，問合せ木は二進木で表わす．根（ルート）から葉（リーフ）へのトップダウン探索とリーフからルートへのボトムアップ探索の両方を可能とするため，木の各ノードは二つの子ノードのポインタと親ノードのポインタを持つ．

前工程で生成された関係演算の並びの先頭から順に１トークン取り出し，ノードを生成して問合せ木に追加する．取り出したトークンが演算子であるなら，生成したノードから下位へとさらに問合せ木を深くしていく．このようにして，式２から問合せ木を組み立てると次のようになる．

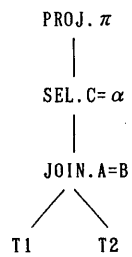

図　３　問合せ木

③単項演算の押し下げ

図３の問合せ木をそのまま解釈して実行すると，表T1とT2を結合した後に，選択，射影を行なうことになる．T1とT2が別のシステムに存在する場合，結合処理のためにT1とT2のすべてのデータを転送しなければならない．一般に，選択，射影はデータ量を削減するものであるので，このような順序で処理を実行すると，結果に現れないデータの転送が生じて，無駄なデータ転送コストがかかってしまう．このように最初から結果に現れないことが分かっているデータの転送を防ぐために，射影や選択などの単項演算子はできるだけデータ転送の前に実行することが望ましい．したがって，前述した問合せ変換により得られた問合せ木から，さらに単項演算子を木のリーフの方向に押し下げる問合せ木変形処理を施す．単項演算押し下げ処理の結果，図３の問合せ木は図４のように変形される．

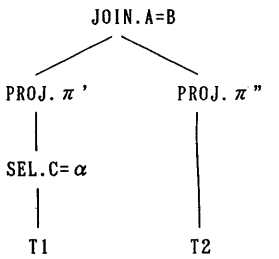

図　４　単項演算押し下げ後の問合せ木

π'とπ"は，図３のπに含まれる属性のうち，それぞれT1，T2に含まれる属性の集合を表わす．また，ここでは属性CはT1に含まれると仮定している．

本システムでは，このような問合せ変換処理と並行して，オペランドに関する情報の収集処理を実行する．その結果，一つのオペランド中のタプルがすべて満たす条件が存在し，それが直上の選択の条件に相反していたら（例えばT1のタプルがすべてC≠αの条件を満たしていたら），その問合せで検索されるタプルは空であるので，結果が空であることを示すコードを返して最適化処理を終了する．

5.3 重複データ情報取得処理

この処理は，利用者ＡＰに指定されたリモート表の重複データがローカルデータベースに存在するかどうか調べ，もし重複データが存在すれば重複部分に関す

る情報を返すものである．重複データの情報は，導出元データベースと導出先データベースで相互に管理しているので，重複データ情報の取得は，リモートデータベースからローカルデータベースに向けて重複データの情報を逐次探索することにより実行される．重複データ情報取得処理は，以下の段階に分かれる．

①探索経路決定

②重複データ情報探索

①探索経路決定

ＡＰに指定されたリモート表から複数のデータベースにデータが導出されている場合，自データベースに関係のある導出データは唯一であるにも関わらず，すべての導出データに対して重複データの探索を行なうことは非常に効率が悪い．このような処理の発散を防止するために，重複データ情報の探索経路を一つに限定する必要がある．探索経路の決定には，3.1で述べたシステム構成表を用いる．システム構成表を用いて，リモートデータベースから自データベースに至る各々のデータベースとその配列順を取り出す．

②重複データ情報探索

導出データの情報は，隣接するデータベース間で相互に管理されている．したがって，隣接データベースの表の重複データ探索にはリモートアクセスは必要ない．垂直分散データベース環境では，局所参照性が存在すると仮定できるので，多くの場合重複データの探索は，ローカルアクセスだけで処理が完了する．しかし，隣接データベース以外のリモートアクセスが全くないわけではない．本システムでは，そのような場合上述した探索経路に基づいて，各データベースの導出元管理表や導出先管理表を検索し，最終的にＡＰに指定されたリモート表と自データベースに存在する表との重複データ情報を得る．図５に重複データ情報探索処理の概念図を示す．重複データ情報探索処理の過程では，導出元管理表または導出先管理表の導出規則に登録されている検索SQL文を取り出し，選択リストを書き換え，検索条件の積をとる．そして，そのような処理を指定されたリモート表が存在するデータベースを起点として，自データベースの方向へ逐次的に実行する．もし，途中で導出されている表が存在しなくなった場合は，重複データが存在しないとして，重複データ情報取得処理は終了する．自データベースに存在する重複データの情報を得ることができれば，その情報を問合せ処理スケジュールの決定に利用する．

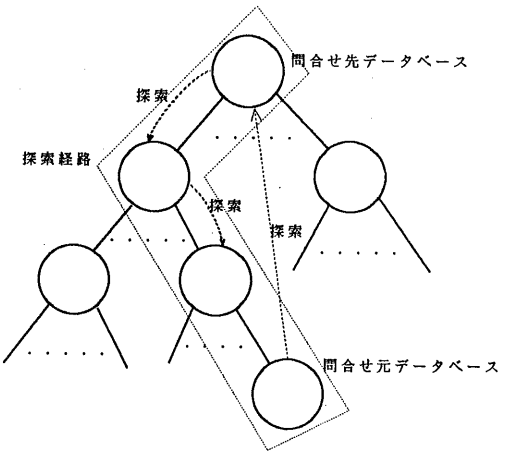

図　5　重複データ探索

5.4 処理スケジュール決定方法

本システムでは，各リモートデータベースでのローカル処理（選択，射影）が終了した後，その結果を自データベースに転送する前に，転送データ量をできるだけ少なくするような処理を行なう．ここでは，いくつかの場合に限定して，適用できる処理スケジュールの決定基準と処理の実現方法について説明する．

①分散結合(join)が含まれる場合

表の結合では，処理の前後で総タプル数が大きく異なることがある．そのような場合，結合処理の結果に含まれないタプルの転送は極力排除して，冗長なデータ転送を防止する必要がある．結果に関連するタプルをあらかじめ特定し，転送データの総量を低減するための手段として，本システムでは準結合(semijoin)を用いる．しかし，すべての場合に準結合が適用されるわけではなく，種々のパラメータを考慮してその適用が可否が決定される（現状では，すべてのパラメータを取り上げ，定量的な判断基準を算出するには至っていない）．ここでは，二変数結合の表の配置にのみ着目して，その処理スケジュールの決定基準について説明する．なお本システムでは，下層のリモートデータベースアクセス機能は，現在実装されている二点間のリモートデータベースアクセス機能をそのまま代用す

るので，リモートデータベースから他のリモートデータベースへ直接データを転送する処理は実行不可能であると仮定する．

・二表が自データベースに存在する場合

この場合はデータ転送の必要はないので，自データベースで結合処理を実行する．

・一方の表がリモートデータベースに存在する場合

この場合は，リモート表をそのまま自データベースに転送する方法と，自データベースの表の結合キーをリモートデータベースに転送し，準結合した結果を自データベースに転送する方法が考えられる．どちらの方法を選択するかは，表のタプル数と結合キーとなる属性の値の種類数によって決定される．リモート表のタプル数をT1，タプルサイズをS1とし，二つの表の結合キーの値の種類数をそれぞれI1，I2，自データベースの結合キーのサイズをsとする．リモートデータベースの表をそのまま自データベースに転送する場合の転送データ量はT1S1である．次に，準結合を用いる場合の転送データ量を算出すると，結合キーのデータ量はI2s，一般的に準結合後のタプル数は，

$$T1I2/\max(I1,I2)$$

と推測できる[3]ので，

$$T1I2S1/\max(I1,I2) + I2s$$

となる．したがって，

$$T1S1 \leqq T1I2S1/\max(I1,I2) + I2s$$

となる場合には基本的にリモートデータベースの表をそのまま自データベースに転送する方法を採用する．

・二表が同じリモートデータベースに存在する場合

この場合は，先にリモートデータベースで結合処理を行ってから結果を自データベースに転送する方法と，両方の表をそれぞれの結合キーで準結合してから自データベースに転送する方法の二つが考えられる．本システムでは，準結合の利用と合わせてローカルデータベースに存在する重複データも利用して（後述），転送コストを低減させる．結合処理を行った後に転送する場合，重複部分を特定する処理が複雑になる．したがって本システムでは，後者のそれぞれに準結合を施した後に自データベースへの転送を行なう方法を採用する．

・二表が別のリモートデータベースに存在する場合

この場合は，表をそのまま自データベースに転送して結合処理を行なう方法と，それぞれに準結合を施した後に自データベースに転送する場合の二つの方法が考えられる．現在，実装されているリモートデータベースアクセス機能には，リモートデータベースから別のリモートデータベースに直接データを転送することはできないので，後者を実現するには，一旦両方の結合キーを自データベースに転送し，両結合キーの結合結果をそれぞれのリモートデータベースに転送して，準結合を行なう必要がある．それぞれの表のタプル数をT1，T2，タプルサイズをS1，S2，結合キーの値の種類数をI1，I2，結合キーのサイズをsとすると，前者の場合の転送データ量は(T1S1+T2S2)となる．後者の場合は，結合キーの転送でそれぞれI1s，I2sのデータ転送必要である．結合キー同士の結合の結果のタプル数は，

$$I1I2/\max(I1,I2)$$

となると推測できるので，結合キー同士の結合結果の転送には，

$$2I1I2s/\max(I1,I2)$$

のデータ転送が必要となる．さらに各リモートデータベースでの準結合の結果のタプル数はそれぞれ，

$$T1I2/\max(I1,I2)$$

$$T2I1/\max(I1,I2)$$

となるので，この方法を用いる場合の転送データ量は，

$$T1I2S1/\max(I1,I2)+T2I1S2/\max(I1,I2)$$
$$+2I1I2s/\max(I1,I2)+(I1+I2)s \quad （式３）$$

と推測できる．したがって，(T1S1+T2S2)が式３の値以下であれば，準結合を用いず表をそのまま自データベースに転送する方法を選択する．

準結合を用いた最適化の場合，本当に最適な解を求めるためには，転送データ量の他に準結合ための内部処理のコストも考慮に入れる必要がある．したがって，上述した計算式から算出された値の単純な比較だけでは不十分である．ここであげた判断基準は基本的なもので，比較される転送データ量の間にある程度大きな格差があれば，そのままこの判断基準が有効であるし，あまり差がない場合はより詳細な評価が必要になる．しかし，現時点ではシステムの性能や通信路の速度に関するパラメータを考慮するに至っていないので，本

試作システム完成後に性能評価データを考慮して，より正確な判断基準を与えることができるように修正可能な構成にしてある．

②重複データの利用

リモートデータベースのデータの部分コピー（重複データ）を自データベースに作成する主な目的は，頻繁に発生するリモートアクセスをローカルアクセスに切り換えることにある．しかし，重複データ作成以前のＡＰが重複データを利用するためには，表指定部分をリモートデータベースのものからローカルデータベースのものへと変換しなければならない．これでは，重複データの利点が十分に活用できない．したがって本システムでは，垂直分散データベースの特長である重複データを利用者が意識しなくても利用できるような機能を持たせている．

前節5.3で説明した重複データ探索処理で得られた重複データ情報を用いて，リモート表と重複部分を持った表を自データベースから探し出す．そして，リモートデータベースにある転送直前の表に対して，自データベースにないデータだけをリモート検索して，自データベースにある重複データをローカルに検索する．そして，それらの実行結果を統合し，自データベースに一時表を作成する．以下で，検索命令作成と検索結果統合の処理について説明する．

図６のような重複データを仮定する．この場合，重複部分を示すSQL文（導出規則）は，

T2: SELECT a,A FROM T1 WHERE C1;（式４）

となる．aは主キー属性，Aは属性の集合，T1は表名，C1は検索条件を表わす式である．このような重複データが自データベースに存在する状態で，ＡＰから，

SELECT * FROM T1 WHERE C2;　　（式５）

という検索要求が出されたとする．検索されるべきデータは，図６の(D1+D2+D3)の部分である．しかし，D1の部分は自データベースに存在するので，システム側でD1を自データベースから，残りのD2，D3をリモートデータベースから検索するようにする．以下に，検索命令作成と検索結果統合の方法を示す．

・検索条件に含まれる属性がすべて重複データに含まれる場合

例で示すと，式５のC2の条件に含まれる属性がすべてaかAに含まれている場合である．D1部分の検索命令の作り方は以下のようになる．

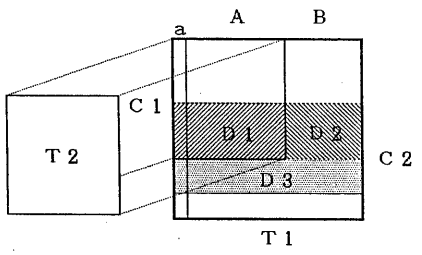

図 6　ローカルデータ検索の概念図

選択リスト……導出規則と利用者SQL文の共通部分

表名……重複データの表

検索条件……導出規則と利用者SQL文の積

D2部分の検索命令の作り方は以下のようになる．

選択リスト……利用者SQL文にあって導出規則にないものと主キー属性

表名……ＡＰで指定された表

検索条件……導出規則と利用者SQL文の積

D3部分の検索命令の作り方は以下のようになる．

選択リスト……ＡＰで指定されたもの

表名……ＡＰで指定された表

検索条件……導出規則の否定と利用者SQL文の積

図６の例で示すと，それぞれの検索命令は次のようになる．

```
D1: SELECT a,A FROM T2 WHERE C1 & C2;
D2: SELECT a,B FROM T1 WHERE C1 & C2;
D3: SELECT * FROM T1 WHERE (NOT C1)& C2;
```

D1とD2の結果の各タプルの対応をとるために，主キー属性aを結合キーとして結合し，その結果とD3の併合をとる．統合処理をSQL文で表わすと以下のようになる．

```
SELECT a,A,B FROM D1,D2 WHERE D1.a=D2.a
UNION
SELECT * FROM D3;
```

・検索条件に重複データに含まれない属性が存在する場合

この場合は，利用者SQL文に示された検索条件を用いて自データベースの重複データD1を検索することはできない．したがって，D1の検索は行わず，D2との結合処理によってT2からD1に対応するタプルを特定する．D2，D3の検索方法は前述した方法と同様で，統合処理

の際に以下のようなSQL文を生成する．

```
SELECT a,A,B FROM T2,D2 WHERE T2.a=D2.a
UNION
SELECT * FROM D3;
```

このようにして，重複部分をSQL文で指定できるような重複データについては，任意の部分重複に対して自データベースにある重複データを利用することにより，リモート検索時の転送データ量が削減できる．

6．結　論

本稿では，現在の社会あるいは企業内で広く行われている業務に対し，その情報の利用形態に適したデータ管理の概念を定義した．そして，それらの業務に適応できる分散データベース管理システムを「垂直分散データベース管理システム」として，現在，それが持つべき基本的な機能を実装した試作システムを開発中である．

本システムの特長は，分散データベースを構成する各ローカルデータベース間の重複データを管理，利用するところにある．重複データ情報の管理では，分散データベース環境に潜在する局所参照性を考慮にいれ，導出元データベースと導出先データベースの双方で重複データ情報を格納することにより，処理の集中や管理情報更新のオーバーヘッドを極力抑制している．検索処理では，リモートデータベースの表の検索に対し，データ転送を行なう前に，最終結果に現れないデータを極力削除し，さらに問合せ発生データベースに存在する重複データを利用して転送データ量を削減するような問合せ最適化処理を実行する．今回実装する重複データ利用方式は，各重複データが元のデータの主キー属性を含んでいれば，任意の検索SQL文で示される部分重複データに適用できる．

どのような問合せ最適化も，すべての問合せに対して十分な効果を発揮するわけではない．現段階では，効果を発揮する条件を与える基準値を算出するには至っていない．試作システム完成後は，今回実装した各機能を実証評価し，問合せ最適化機能の使用を決定するためのパラメータと決定基準を求める必要がある．また，さらに管理情報の一時保管や経験的知識の獲得などの機能をつけ加え，より最適な処理を選択できるシステムを構築していく予定である．

なお，本稿では，本システムに実装する機能の中で，特に導出データ定義機能と問合せ最適化機能について説明した．今回説明はしていないが，重複データの更新機能も開発中である．

謝　辞

本研究開発は，通産省工業技術院大型プロジェクト，「電子計算機相互運用データベースシステムの研究開発」の一環として，日本電気㈱がＮＥＤＯから委託を受けて実施したものである．

なお，本研究内容に関して貴重な御意見を頂いた浜館昌樹，稲葉亜矢子両氏をはじめとする日本電気㈱基本ソフトウェア開発本部の皆様に深く感謝致します．

文　献

[1]鶴岡，高野「垂直分散データベースのモデルと基本機能について」，情処データベース・システム研究会研報，Vol.90,No.36，77-2，1990.5.17

[2]高野，鶴岡「垂直分散データベースにおける重複データの処理について」，第42回情処全国大会講演論文集(4)，pp.131-132，1991.3.12-14

[3]Jeffrey D. Ullman,"PRINCIPLES OF DATABASE AND KNOWLEDGE-BASE SYSTEMS Vol.Ⅱ", COMPUTER SCIENCE PRESS