1H−10

SQLコンパイル方式を用いた分散DBMSのレコード転送制御方式

赤間 浩樹　梅本 佳宏　中村 仁之輔
NTT情報通信網研究所

1．はじめに

近年、公衆網の高機能化が進められている[1]。特にフリーダイアルやダイアルQ²などの高度通信処理サービスでは、今後のサービス追加や顧客増に柔軟に対応させるためにネットワーク上のリレーショナル型DBの実体を水平分割させたいという要求がある。その分散DB構成のイメージを図1に示す。

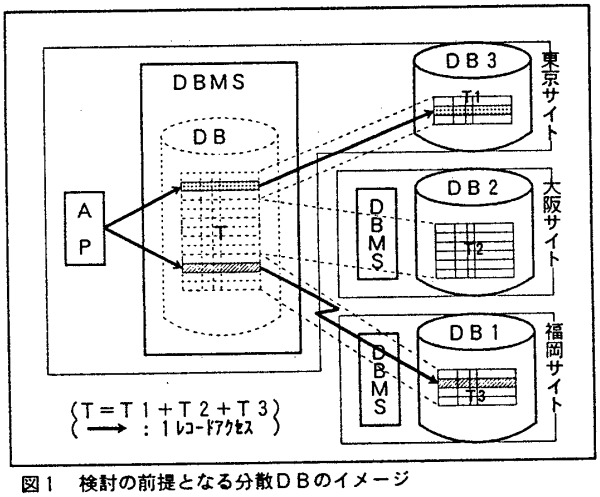

図1　検討の前提となる分散DBのイメージ

このような高度通信処理サービスで求められるシステムにおけるDB、および、DBアクセスの特徴をまとめると次のようになる。

（1）DBは複数のサイト間で水平分割されている。つまり、各サイトのデータ構造は同一で、内容が異なるデータを各サイトに持つ。
（2）中心業務は1レコードアクセスのショート・トランザクションであり、高速性が重視される。また、アクセス頻度も非常に多い。
（3）運用情報収集のための複数レコードアクセスも必要とされる。しかし、アクセス頻度は少ない。

2．アクセス高速化の課題

これまでの分散DBMSにおける問合せの高速化の研究は、結合処理（JOIN）のような高度な処理に関する最適化が中心であった。しかし、本検討が対象とするシステムではそのような高度な処理の頻度は非常に少なく、逆に、単純なアクセスが多い。

このようなシステムでは、分散DBMSを採用することによる処理速度の低下を避けるため、他サイトへのアクセスの基本となる次の3つの処理の高速化が重要になる。

① サイトの選択処理
② サイト間のレコード転送処理
③ 複数サイトからの結果のマージ処理

そこで本稿では、その中で最も処理時間のかかる②の処理についてシミュレーションの結果を示し、レコード転送の実現方法を提案する。

3．レコード転送方式の分類とシミュレーション

分散DBMSでは問合せを複数のDBMSの動作によって実現する。そこで、APのある側のDBMSをクライアント、DBのある側のDBMSをサーバと呼ぶことにする。クライアントとサーバ間のレコード転送方法として、以下のような方法を考えることができる。

【方式1】1レコード毎に転送する方式
【方式2】複数レコードをまとめたブロック毎に転送する方式
【方式3】該当する全てのレコードを一括転送する方式

これらについて以下のパラメータを用いてシミュレーションを行った。

（a）問合せパターン（検索系）
　① オンライン業務の中心である1レコード検索
　② 運用情報収集等に用いられるソートなしの単純な複数レコード検索
　③ 高度なリレーション操作（副照会、JOIN）を必要とする複数レコード検索
（b）方式3におけるブロックサイズ
　① 1ブロックに100レコード……方式3a
　② 1ブロックに10レコード ……方式3b
（c）複数レコード検索時のレコードfetch回数
　（該当レコード1000件中）
　① 1レコードfetch
　② 100レコードfetch

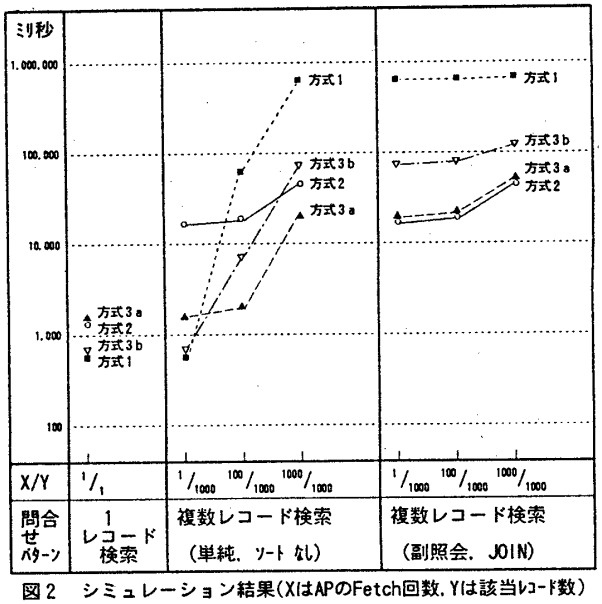

図2　シミュレーション結果(XはAPのFetch回数, Yは該当ﾚｺｰﾄﾞ数)

Record transmission method in distributed DBMS
Hiroki AKAMA, Yoshihiro UMEMOTO, Jinnosuke NAKAMURA
NTT Network Information Systems Laboratories

③ 全レコードfetch

この結果を図２に示す。

４．結果

今回のシミュレーションより、次の結果を得た。

（１）１レコード検索では、１レコード毎に転送する方式１が最もよい。方式１は方式２や方式３ａに比べ転送時間が約半分になる。また、方式３ｂに比べても方式１が10%以上よい。

（２）ソートのない単純な複数レコード検索の場合には、fetchする回数、および方式３のブロックサイズによって最適な方式が変化する。

① fetchする回数が少ない場合には方式１が最も高速である。しかし、fetchする回数が増えるに従って急激に悪くなる。方式２は逆にfetchするレコードが少ない場合に悪い。方式３は中間の性質を持つ。

② 方式３はブロックサイズにより性能が変化する。ブロックサイズを短め(10)にした場合には、方式２より倍以上遅くなり、ブロックサイズを長め(100)にした場合には方式２の半分程度になる。

よって、方式３ａの方式が１レコードfetchでは方式１の２倍程度かかるものの、全体を見た場合に高速である範囲が最も広いので、この問合わせパターンに適当と考えることができる。

（３）副照会やJOINを行う複数レコード検索の場合には、サーバ側のデータの全てをいったんクライアント側に転送する必要がある。従ってfetchする回数によらず方式２が最もよい。

また、図２では条件に該当するレコード数は1,000レコードとしたが、10,000レコードでもこれらの傾向は変わらなかった。以上から次の結論を得た。

【結論】 表１に示すように、SQLの問合せパターンによってクライアントとサーバ間のレコードの転送方式を変えることが有効である。

表１ 評価結果のまとめ

	問合せパターン	最適方式
1	１レコード検索	方式１
2	複数レコード検索（単純、ソートなし）	方式３
3	複数レコード検索（副照会、JOIN）	方式２

５．実現方法

この結論を最適化として実現するための手法について示す。なお、本稿で検討の前提とするDBMSはSQLコンパイル方式を採用しているものとする。なぜなら、オンライン・トランザクションに対応するために、高速なSQLの実行が必要とされるからである。

本最適化を実現するには、問合せに対する最適なレコード転送方式の判定処理と切り替え処理が必要になる。しかし、これを実行時に行ったのでは最適化の効果（特に１レコードアクセスの高速性）が減ってしまう。そこで、SQLコンパイル方式を採るDBMSの特性を活かし、これらの判定処理もプリコンパイル時に行う方法を示す。

（１）プリコンパイル時にSQLのパターン判定を行いレコード転送方法を決定する。（図３ａ）

（２）そのレコード転送方法の実現ロジックを、クライアント側のロジックはCM（クライアント側アクセスモジュール）中に、サーバ側のロジックはSM（サーバ側アクセスモジュール）中に作成し、それぞれのDBMSのシステムDB中に登録しておく。同時に、そのCMとSMの番号をもとのAP中に埋め込んでおく。

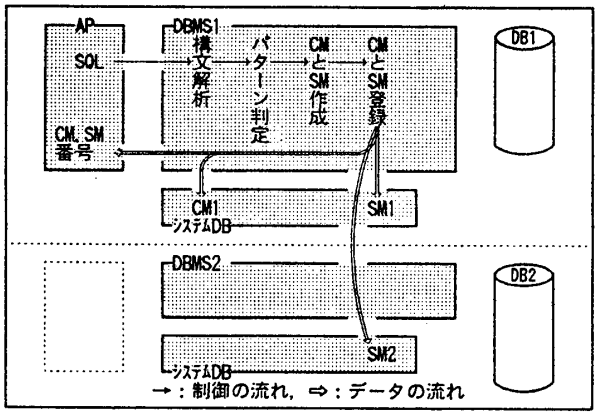

図３ａ プリコンパイル時の処理イメージ

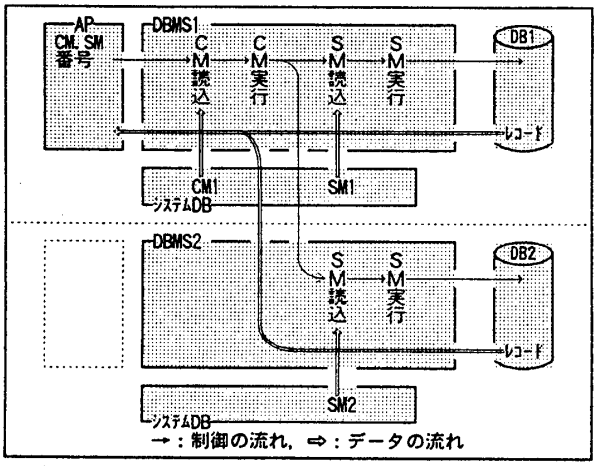

図３ｂ 実行時の処理イメージ

（３）APをコンパイルする。

（４）APの実行時には、AP内のCM,SM番号に該当するCMとSMを読込み、その中に書かれているレコード転送ロジックを実行する。（図３ｂ）

このようにすれば、AP実行時の負荷を全く増加することなくレコード転送方法の判定および切り替えが実現できる。

６．まとめ

本稿では、水平分割されたDBを持つリレーショナル型分散DBMSにおいて、クライアント側のDBMSとサーバ側のDBMSのレコード転送方法に着目したシミュレーションの結果を示した。

そして、SQLの問合せ毎に最適なレコード転送方法が変化するという結果から、SQLパターンによってレコード転送方法を切り替えるという最適化手法の有効性を示し、SQLコンパイル方式を採るDBMSで実現する場合のイメージを示した。

今後は、検索系以外の検討、およびＳＱＬパターンの詳細化、レコード転送方法の詳細化を行っていきたい。

参考文献

[1] 秋山稔 他：インテリジェントネットワークとネットワークオペレーション，コロナ社，1991