

6X-5 Web上に分散したデータベースへの問い合わせ最適化手法

一色 誠司[†] 横山 昌平[†] 太田 学[‡] 石川 博[‡]

[†] 東京都立大学工学部電子情報工学科

[‡] 東京都立大学大学院工学研究科

1. はじめに

現在、多くの商用サイトが関係データベース(RDB)を用いたWebページを構築している。このようなサイトではWebサーバに併設した単一のDBに対してアクセスするのが一般的である。一方、e-commerce等の普及に伴い、ネットワーク上に存在する複数のDBサイトからデータを動的に結合してブラウザに表示するといった必要性が生じてくる。しかし、既存のRDBは、複数のDBサイトにまたがるjoinを許していない。そこで、リモートDBが管理下になく一時テーブル等が持てないという条件下で、データを動的にjoinする手法と最適化について検討を行った。

2. 最適化手法

ネットワークを介してjoinを行うために、リモートDBに対して問い合わせを行い、検索結果をローカルDBに一度格納(コピー)したのち、ローカルDBに問い合わせ(join)を行うという方法を用いた。この流れをFig. 1に示す。このときの処理時間はネットワーク間のデータ送受信量と接続の確立時間に依存する。データ量を減らすためには、join

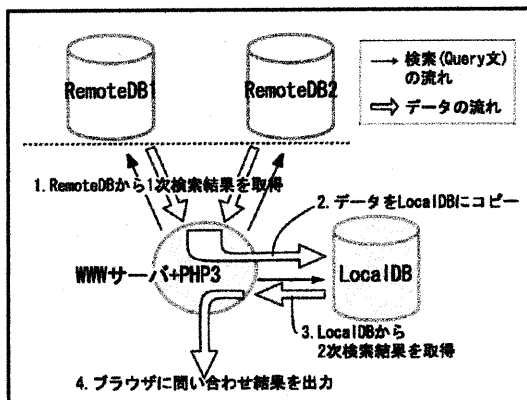


Fig. 1 システム概要図

Query optimization method for distributed databases on the Web.
Seiji Isshiki[†], Shohei Yokoyama[†], Manabu Ohta[‡] and Hiroshi Ishikawa[‡]
[†] Faculty of Engineering, Tokyo Metropolitan University
[‡] Graduate School of Engineering, Tokyo Metropolitan University

の結果に必要なデータを何らかの方法によって発見し、そのデータのみを取得すればよい。

ここでは、semi-join[1]、min-max[2]という二つの手法を前述の分散環境に適応出来るように発展させ、これを用いて予備問い合わせを行い、不要なデータを取り込まないようにした。以下に、二つの手法について述べる。

2.1 semi-join

この手法は予備問い合わせでjoinのキーカラムのみを取得してjoinを行うことによりタプル数を限定し、再度要求を満たすカラムをすべて取得する方法である。joinのアルゴリズムには、複数のDBのデータをすべて取得し同等に扱ってjoinする方法(並列的手法 Fig.2)と、データを取得するたびに

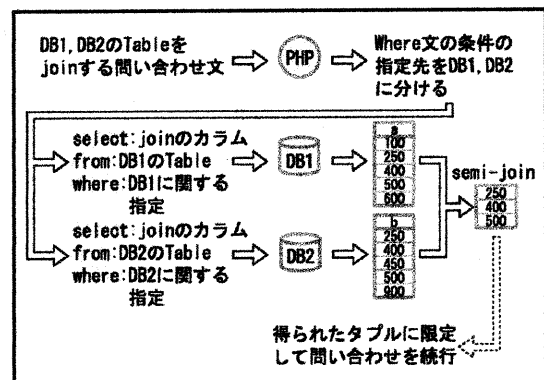


Fig.2 semi-join(並列的手法)の例

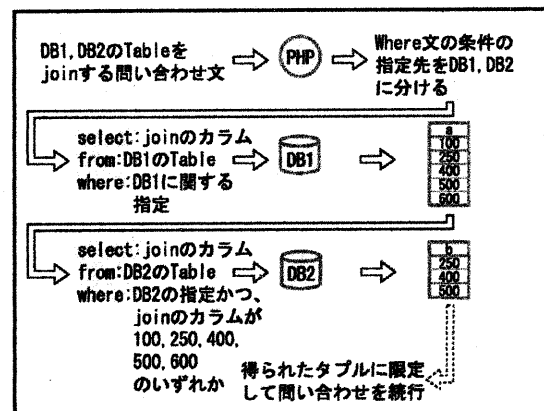


Fig.3 semi-join(直列的手法)の例

前のデータと比較し絞り込むことで join する方法 (直列的手法: Fig.3) を考案した。

2.2 min-max

この手法は、Fig.4 に示すように、予備問い合わせで join のキーフィールドの最小値、最大値を取得することを繰り返し、取得のたびに前のデータと比較し絞り込むことでタプルを限定する方法である。

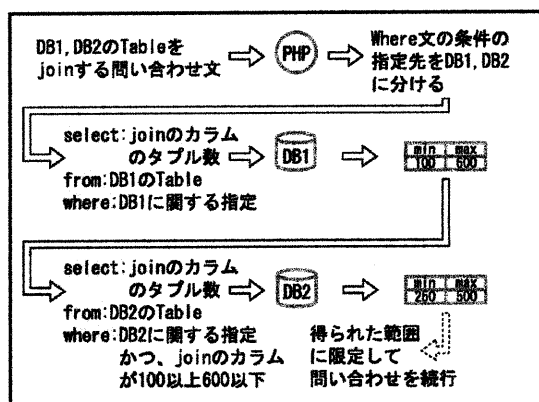


Fig. 4 min-max の例

3. 実験

PC-UNIX 上で、PHP3 と PostgreSQL6.5 を用いて実験を行った。また、商品の ID、内容及び説明からなるテーブル (944 件) と商品の ID、価格及びポイントからなるテーブル (1556 件) を用意した。

実験には、join 指定を除く where 句の中に、(1) 条件に等号を含む場合 (検索結果: 1 件)、(2) 条件に不等号を含む場合 (検索結果: 65 件) を指定した 2 つの問い合わせを用いた。それぞれのアルゴリズムに対する実行時間を Fig. 5 に示す。

問い合わせ(1)、(2)いずれの場合も最適化手法を適用することにより実行時間が短縮することが確かめられた。semi-join は並列より直列手法が優れていること、min-max 単独では semi-join に比べそれほどの効果が期待できないことが確かめられた。そこで semi-join(直列)と min-max を併用したところ、(1) では実行時間がわずかに増加したが、(2) では値が改善された。これは、(1) では等号を含む条件によってデータが既に絞り込まれるため、min-max は効果を発揮せず、逆に最小値、最大値を得るための前処理によるオーバーヘッドが現れた結果だと考えられる。このことから、等号を含む条件が指定されている場合には、semi-join のみを適用するアルゴリズムが効果的である。

	問合せ(1)	問合せ(2)
最適化アルゴリズムなし	4.354	4.602
semi-join(並列)	1.062	1.997
semi-join(直列)	0.932	1.904
min-max	3.606	3.565
semi-join(直列)+min-max	0.962	1.524

単位: sec

Fig. 5 実験結果

4. おわりに

本稿では、複数 DB に対して問い合わせを発行できる問い合わせ言語があった時、その問い合わせを既存の SQL に分解し最適化を行うアルゴリズムを提案した。データの入出力形式はインターフェイスに依存しないよう考慮した。検索エンジンの検索結果のように、一度に表示する件数が制限されている場合には、さらなる最適化が可能である。

さらに現在では、XML 文書を DB として利用する研究が盛んに行われている。既存の DB は DBMS によって管理されているが、XML では必ずしもその必要がない。XML 文書は Web との親和性が高く、ファイルを Web 上に公開するだけでも有用な DB となり得る。これまでデータの提供業者は、データだけでなく DBMS やそれにアクセスするための手段を用意してきた。しかし、これからは DB そのものをサービスとして提供する DB サプライヤーといったビジネスが生まれるであろう。本研究はそのためのステップとなり得るものであり、今後 XML の問い合わせ言語に対し同様のアプローチをすることを検討している。

参考文献

- [1] F. P. Palermo.: A Data Base Search Problem.: In Information Systems: COINS IV(ed., J.T. Tou).New York, N.Y.: Plenum Press(1974).
- [2] A. Makinouchi, M. Tezuka, H. Kitakami, and S. Adachi. "The Optimization Strategy for Query Evaluation in RDB/V1." Proc. 7th International Conference on Very Large Data Bases, Cannes, France (September 1981).