

ネットワーク間異種 DB アクセス最適化方式

4Z A-06

熊谷 昌大† 清水 晃‡ 広島 清美† 嶋崎 康一†  
 † (株) 日立製作所ソフトウェア事業部  
 ‡ (株) 日立製作所中央研究所

1. はじめに

データベースハブは、データベース管理システム(以下 DBMS)の論理的な統合を可能にする情報提供基盤であり、DBMS の存在する環境や種類を意識せずにネットワークを介した「異種 DB アクセス」を実現するものである。必要なデータを各 DBMS のインタフェースを個別に意識することなく取り出し、一つのインタフェースで参照できることが特徴であるため、ユーザは、アプリケーション開発の効率化、新規業務の迅速な立ち上げをおこなうことができ、予期しない市場の急激な変化にも低コストで柔軟に対応することが可能となる。本論文では、データベースハブによる異種 DB アクセスの最適化方式について報告する。

2. 異種 DB アクセス

異種 DB アクセスは、様々な環境に散在する DBMS(以下外部 DBMS)に対して SQL を発行し、ネットワークを介してデータを取得する。そのため、DBMS 間のデータ通信が検索性能のネックとなることが多い。DBMS 間のデータ通信量は探索条件評価のタイミングを含めた DB アクセスの順序(以下プラン)によって大きく変化するため、異種 DB アクセス(特に複数の DB にアクセスする結合や副問合せを処理する場合)では、データ通信量を小さくするプランを選択することが高い検索性能を得るために有効である。そこで、以下の二つの観点から異種 DB アクセスプランを選択する。

- ・ 同一 DBMS の複数表にアクセスする時は、ひとつの SQL にまとめ上げ DBMS 間の通信回数を減らす。(以下戦略 1)
- ・ 探索条件を可能な限り外部 DBMS 上で評価することで、DBMS 間のデータ通信量を小さくする。(以下戦略 2)

3. 異種 DB アクセスプラン選択時の課題

上記二つの観点は、異種 DB アクセスを要する多くの SQL において有効であるが、次に示すような SQL を処理する場合に問題となることがある。

3.1 同一 DBMS 表まとめ上げ(戦略 1)の問題

```
select A1.c1 from A1, A2, B1 where A1.c1=A2.c1
and A1.c2=B1.c2 and B1.c3=0
(A1,A2はDBMS1の表、B1はDBMS2の表とする)
```

図 1 SQL1

ユーザが図 1 に示す SQL(以下 SQL1)を発行した場合を考える。データベースハブで本 SQL を実行するにあたっては、図 2 に示す二つのプランが考えられる。

同一 DBMS に存在する A1,A2 に対する SQL をまとめ上げて A1-A2 間の結合を DBMS1 上でおこなうという戦略 1 を考慮した図 2(a)のプランは、DBMS1 における

SQL optimization for multi DB access through network

† Shota Kumagai, ‡ Akira Shimizu,  
 † Kiyomi Hirohata, † Kohichi Shimazaki  
 † Hitachi,Ltd.,Software Division,  
 ‡ Hitachi,Ltd.,Central Research Laboratory

A1-A2 間の結合結果行数により、DBMS 間のデータ通信量が小さくなり、高い検索性能を得ることができる可能性と、逆に 2 段目の結合で、データの突合せ回数が多くなり、検索性能が悪くなる可能性がある。

また一方で、A1-B1 の結合を最初におこなう図 2(b)のプランでは、DBMS 間のデータ通信量が大きくなる反面、B1.C3=0 という探索条件があるため A1-B1 の結合結果数は小さくなり、2 段目の結合処理が高い性能でおこなわれる可能性がある。DBMS1 が存在するマシンの性能が、ハブとなる DBMS が存在するマシンの性能に比べて大きく劣る場合は、このプランは特に有効である。

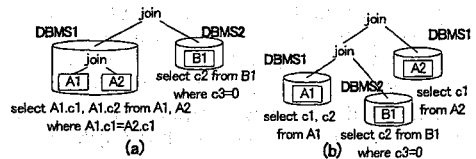


図 2 SQL1 に対するプラン

このように、単純に戦略 1 を採用すると処理時間の大きい結合や副問合せを外部 DBMS で処理させることになり、検索性能が悪くなる可能性があるという問題がある。この問題を解決するためには、プラン選択時に、外部 DBMS の処理能力と、DBMS 間の通信コストを考慮する必要がある。

3.2 外部 DBMS 上での絞り込み(戦略 2)の問題

また、ユーザが図 3 に示す SQL(以下 SQL2)を発行した場合を考える。

```
select A1.c1 from A1, A2, B1 where A1.c1=B1.c1
and A2.c1=B1.c1 and B1.c1=0
(A1,A2はDBMS1の表、B1はDBMS2の表とする)
```

図 3 SQL2

データベースハブで本 SQL を処理する場合、A1,A2 をまとめ上げてしまうと、DBMS1 において A1 と A2 の直積がおこなわれデータ量が増大し、DBMS 間のデータ通信量および、2 段目の結合処理時間が大きくなってしまふ(図 4(a))。そのため、探索条件によりデータを絞り込むことができると思われる B1 を先に結合する戦略 2 を考慮したプラン(図 4(b))が有効であると考える。

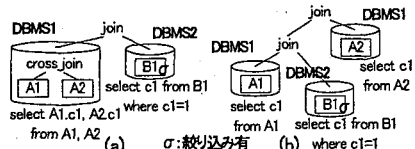


図 4 SQL2 に対するプラン

しかしながら、SQL2 では探索条件から B1.C1 と A1.C1 の値はそれぞれ A2.C1 の値と等しいことがわかるため、A1.C1=A2.C1 AND A1.C1=1 AND A2.C1=1 という条件を導き出すことができる。これらの条件を探索条件として新たに生成することにより図 4(a)のプランと同じ手順で、絞り込

みの強い表同士の結合を DBMS1 においておこなうことができる図5に示すプランとすることができる。

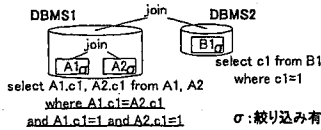


図5 探索条件の生成

このように、戦略2を採用しようと思っても、ユーザの書くSQLが、必ずしも、各外部DBMSに閉じた条件を切り出した形になっていないため有効なプランを得ることができない場合がある。このため、SQLの探索条件を変形してDBMSに閉じた新たな探索条件を生成する機構の実装が必要となる。

4. 解決手段

上記異種DBアクセスプラン選択時の問題を解決する手段を以下に示す。

4.1 絞り込みを考慮した最適化

近年、多くのDBMSは、ユーザによるSQLチューニングの負担を最小限にするため、プラン選択時にコストベースのSQL最適化方式[1]を採用している。ユーザが発行するSQLが異種DBアクセスを必要とするものである場合、各DBMSでの検索結果行数、各DBMSが動作しているマシンやそれらを繋ぐネットワークの性能を考慮してSQLの実行コストを算出し、コストベース最適化を適用することを考える。

各DBMSでの検索結果行数は、SQL文中に指定された単表に閉じる探索条件や、結合列の情報を考慮して予測する。また、各DBMSの存在するマシン性能およびネットワーク性能は、次に示すようなパラメタの値より算出することが可能である。各パラメタは実運用環境を基にユーザが設定可能とする。

- ・ ネットワーク通信スループット
- ・ ネットワーク通信遅延時間
- ・ 外部DBMSが存在するマシンの性能

SQL1のように、それだけでは最適なプランの判断が困難なSQLを処理する場合、このような要素を組み合わせ、外部DBMSにおける絞り込みがプラン全体に与える影響を予測し、実行コストを算出することで有効なプランを選択することが可能となる。

4.2 探索条件の生成

次に、SQL2のように、あるDBMSに閉じる探索条件がないためにDBMS間のデータ通信量が大きくなる恐れがあるSQLについて、データ通信量を減らすことで高速に処理することを目的として新たな探索条件を生成することを考える。

まず、SQLに指定された探索条件に対してCNF変換をおこなう。CNF変換とは、図6に示すようにORで結ばれた探索条件をANDで結ばれた形式へ変換することである。変換後、外部DBMSにおける絞り込みを強くする条件を元の条件に付加することでデータ通信量を小さくする。

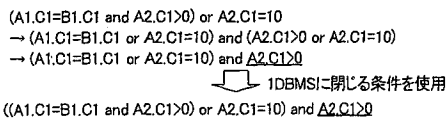


図6 CNF変換による探索条件の生成

さらに、探索条件を推移させて、新たな探索条件の生成

をおこなう。条件の推移による探索条件生成処理の考え方は以下の通りである。SQL2を処理する場合の例を図7に示す。

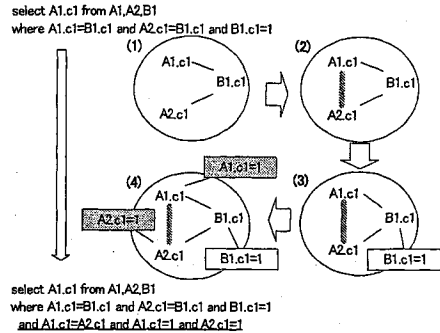


図7 条件推移による探索条件の生成

- (1) SQLの中の探索条件の中から「列指定=列指定」となっているものを抜き出し、列をノード、=をエッジとしたグラフを生成する。グラフは、1つまたは複数個になる。
- (2) (1)で生成した各グラフについて、グラフ内でまだ結合されていないノード間をエッジで結ぶ。このエッジが新たな探索条件である。
- (3) SQL中の単表に閉じる探索条件を(2)のグラフのノードに付与する。
- (4) エッジで結ばれた全てのノードに新たな探索条件として同一条件を付与する。

ここで、新しい条件の生成が検索性能向上の観点で必要ない場合や、生成したことでかえって性能劣化を招く可能性がある場合には、探索条件を生成しない。生成しない探索条件を以下に示す。

- ・ 副問合せ内の探索条件
- ・ 異なるDBMSにまたがる探索条件
- ・ ユーザによる探索条件の生成抑止指定がある場合

5. 効果

それぞれ10,000行のデータを格納した、A1, A2, B1の3表の結合をおこなう問合せで、B1の選択率が0.01、検索結果が100行となるケースにおいて、SQL1で、外部DBMSにおける絞り込みの考慮を優先した図2(b)のプランは、同一DBMSに対するSQLのまとめ上げをおこなう図2(a)のプランの1.1倍の性能を得ることができた(図8(a))。また同様のケースでSQL2では、新たな探索条件を生成し、DBMS間のデータ通信量を小さくした図5のプランが図4(a)のプランの1.5倍の性能を得ることができた(図8(b))。

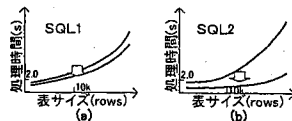


図8 効果

6. おわりに

本論文では、ネットワークで結ばれた複数のDBMSを論理的に統合するデータベースハブにおいて有効な各DBMSの処理性能、検索結果行数を意識したアクセス最適化方式および効果について報告した。

7. 参考文献

[1] P. Griffiths Selinger, et al.: "Access Path Selection in a Relational Database Management System", In Proc. Inter. Conf. Management of Data(ACM)(Boston, Mass., May 1979), 23-34.