2Q-7 パイプライニングによるサイト間結合演算の並列処理

斎藤邦子、坂本明史、川上　英、疋田定幸
沖電気工業（株）

1. はじめに

質問処理において、結合演算はコストの高い処理を要する。分散データベースで、結合する表が異なるサイトにある場合、通信のオーバヘッドも加わるため、効率化は重要である。分散では、複数の計算機による負荷の分担、並列処理の利点がある。従って、通信オーバヘッドの削減と並列性のバランスを考慮した効率化が必要である。

並列処理の有効な方法に、パイプライン処理が提案されており[1]、結合演算処理では、パイプラインネストループ法が有効である[2]。パイプライン処理では、データがスムーズに流れ、処理を効率化するために、1回の通信で送るデータ件数(ブロッキングファクタ)に適当な値を設定することが重要である。本稿では、パイプラインネストループ法の処理において、ブロッキングファクタと並列性,通信オーバヘッドの削減について考察し、最適なブロッキングファクタを示す。

なお、本研究は通商産業省工業技術院の大型プロジェクト「電子計算機相互運用データベースシステムの研究開発」の一環として研究開発を進めているものである。

2. 分散データベースモデル

分散データベースは、ローカルなＤＢＭＳを持つ複数のサイトから構成され、自サイト、他サイトのデータベースアクセスにはＳＱＬの機能が提供されているとする。ＳＱＬは、データのフェッチを１タプル単位に行うものと定義している。しかし、サイト間の結合演算を行う場合、他サイトのデータのフェッチを１タプルづつ行っていたのでは、通信オーバヘッドが大きい。通信回線の高速化により、転送量が通信時間に及ぼす影響は少ないので、他サイトのデータをｎタプル(ブロッキングファクタ)づつフェッチすれば、オーバヘッドの減少が期待できる。以下では、パイプライン処理のためのプリフェッチの機能を示し、2サイト間、3サイト間の結合演算処理を説明する。

2.1 プリフェッチ

サイト１のデータをサイト２が処理する場合、パイプライン処理の目的は、サイト１から２へデータが流れ、サイト１、２の処理が止まらずに並列に行われることである。そのために、非同期通信プリミティブによるプリフェッチの機能が必要である。図１は、プリフェッチによる並列処理を示している。サイト２では、サイト１よりタプルを受け取ったならば、次のタプルの要求をだし(図1①)、応答を待たずに、受け取ったタプルの処理を行う。処理が終了し、次のタプルが必要になったならば、応答メッセージを受け取る(図1②)。

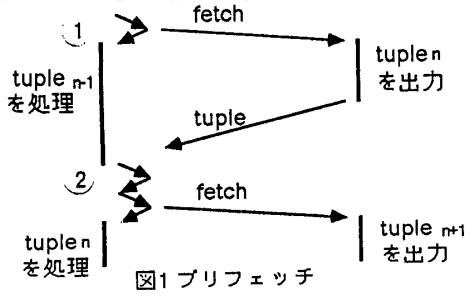

図1 プリフェッチ

2.2 結合演算処理

まず、サイト１，２のリレーション R_1, R_2 の結合の処理を説明する。Bnはサイトｎから転送されるタプルのブロッキングファクタである。サイト１はサイト２の要求により、R_1 から条件なしで B_1個のタプルをフェッチし転送する。サイト２では、まず、R_2の検索のために、R_1の最初のタプルと結合するタプルを得るための条件を付ける。例えば、結合条件が $R_1.A=R_2.A$ の場合、R_2の検索には、"$R_2.A$=パラメータ"の条件が付けられており、パラメータに R_1のタプルの A の値をセットする。R_2のタプルをフェッチし、結合結果を得る。フェッチの結果、"該当タプルなし(End Of Tuple)"になったならば、R_1の次のタプルに対して、同様の処理を行う。サイト２は結合処理を行う前に、R_1の次の B_1個のタプルを得るための要求をだしておく。次に、サイト１，２，３のリレーション R_1, R_2, R_3の結合の処理を説明する。サイト２は、サイト３からの要求により上記の方法で R_1, R_2の結合を行い、B_2個のタプルを得たならば、転送する。サイト３は、R_1, R_2の結合結果とR_3の結合を同様に行う。この場合も、次の R_1, R_2の結合結果の B_2個のタプルのプリフェッチの要求を出しておく。

3 ブロッキングファクタと処理効率

3.1 最適なブロッキングファクタ

効率的にパイプライン処理を行うには、サイト２がサイト１のデータを待つ時間(W_2)を短くしなければならない。図2はサイト１，２の処理を示したものであり、W_2 は次の2つをあわせたものである。

① プリフェッチによる待ち時間
② サイト１から最初のデータが届くまでの待ち時間

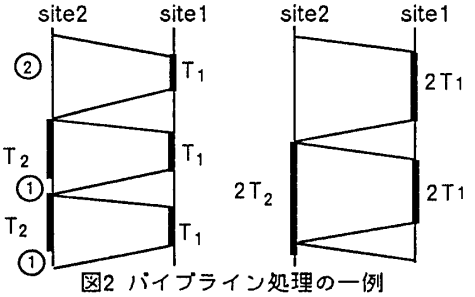

図2 パイプライン処理の一例

①は、サイト２の処理が終わり、次のタプルが必要となったときに、サイト１からタプルが届いていないために生じる待ち時間である。この待ち時間は、通信コストが転送量に依存しないとすると、ブロッキングファクタに適当な値を設定することで、削減できる。図2では、サイト１からサイト２に送るタプル数を２倍にしても、通信時間は変わらないので、１タプルあたりの通信オーバヘッドは1/2になり、処理時間の差に吸収される。従って、次の式を満たす場合、①は生じない。

$Ti_2 - To_1 >= T/B_1$ ・・・・ (1)

To_1: サイト１が、１タプル出力するための時間
Ti_2: サイト２が、受信したデータ、１タプルを処理する時間

Achieving Parallelism in Intersite Join Operation by Pipelining
K. Saito A. Sakamoto S. Kawakami S. Hikita
OKI Electric Industry Co., Ltd.

B_1: サイト1が1度に転送するタプル数
T: 要求とデータの転送のための通信時間(転送時間)

(1)を満たしていない場合、一回の通信で生じる待ち時間は、

$(T + B_1 * To_1) - B_1 * Ti_2 = T + B_1 * (To_1 - Ti_2)$

であり、転送回数は、N_1/B_1

N_1: サイト1が転送する全タプル数

であるので①は、

$(T + B_1 * (To_1 - Ti_2)) * (N_1/B_1)$

になる。

☒は、並列動作に入るまでの待ち時間であり、

$T + B_1 * To_1$

になる。

従って、ブロッキングファクタ(B_1)と全体の待ち時間(W_2)の関係は、

i) 式(1)を満たしている場合(①の待ち時間がない)

$W_2 = T + B_1 * To_1$

ii) 式(1)を満たしていない場合

$W_2 = T + B_1 * To_1 + (T + (B_1 * (To_1 - Ti_2))) * (N_1/B_1)$

である。ブロッキングファクタが大きくなると、通信のオーバヘッドは削減されるが、並列動作に入るまでの待ち時間が増える。上記の式より、W_2を最小にするブロッキングファクタBo_1は、以下のようになる。

i) $Ti_2 > To_1$ かつ $T/(Ti_2-To_1) <= \sqrt{N_1 * T / To_1}$ の場合

$Bo_1 = T/(Ti_2-To_1)$

ii) i) 以外の場合

$Bo_1 = \sqrt{N_1 * T / To_1}$

i)では、通信オーバヘッドは、最初のデータの転送のときだけである。ii)は、これ以上大きな値にすると、通信のオーバヘッドは下がるが、最初の待ち時間が増え、並列性が下がる場合である。

3.2 ネストループ法でのブロッキングファクタ

パイプラインネストループ法を用いた2サイト、3サイト以上の結合での、最適なブロッキングファクタについて考察する。そのためには、データを送るサイトと、受け取って処理するサイトの処理時間を見積らなければならない。各サイトの処理時間は、(フェッチの回数)*(1タプルのフェッチに要する時間)とみなすことができる。リレーションのタプル数、セレクティビティ等を用いて、処理時間を見積もる。セレクティビティは、、結合結果のサイズを見積もるためのものであり、R_1, R_2のセレクティビティを S_{12}とすると結果タプル数は $N_1*N_2*S_{12}$になる。

①2サイト結合

R_1, R_2の結合で、サイト2は、R_1の各タプルについて EOT まで検索するので、フェッチの回数は、(結果タプル数+N_1)となる。これを、N_1で割ったものが、$R_1$1タプルを処理するのに必要なサイト2のフェッチの平均回数であり、$N_2*S_{12}+1$となる。従って、To_1, Ti_2は以下のような見積りになる。

To_1: Cs

Ti_2: $C * (N_2*S_{12} + 1)$

Cs: 条件なしのシーケンシャルなフェッチに要する時間

C: 条件付のフェッチに要する時間

②3サイト以上の結合

R_1, R_2, R_3を結合する場合、サイト2からサイト3へ送るタプルの最適ブロッキングファクタを求めるために、サイト2、3の処理時間の見積りについて述べる。サイト2で、R_1, R_2を結合するためのフェッチの回数は①の通りである。これを結果タプル数で割ったものが、R_1, R_2の結合結果1タプルを得るために必要なフェッチの平均回数であり、$1 + 1/(N_2 * S_{12})$となる。サイト3が R_1, R_2の結合結果1タプルを処理するために必要な時間は、①の Ti_2と同様に見積もることができる。従って、To_2, Ti_3は以下のようになる。nサイトの結合の場合も同様に見積もることができる。

To_2: $C * (1 + 1/(N_2 * S_{12}))$

Ti_3: $C * (N_3 * S_{23} + 1)$

以上の見積りで得た最適ブロッキングファクタを、転送時間、タプル数等に適当な値を設定した例で示す。

例) データベースアクセス時間を Cs=20msec, C=40msec、リレーションのタプル数を N_1=10000, N_2=12000, N_3=11000、セレクティビティを S_{12}=1/12000, S_{23}=1/10000 とする。転送時間(T)が、80msec, 160msec, 800msec の場合について示す。800msec は、衛星通信のように通信時間がかかる場合を想定している。

R_1, R_2の結合では、To_1=20msec, Ti_2=80msec となり、

T=80msec の場合 Bo_1=2、

T=160msec の場合 Bo_1=3

T=800msec の場合 Bo_1=14 である。

R_1, R_2, R_3の結合では、To_2=80msec, Ti_3=84msec となり、

T=80msec の場合 Bo_2=20、

T=160msec の場合 Bo_2=40、

T=800msec の場合 Bo_2=200 である。

以下のグラフは、サイト2から転送するタプルのブロッキングファクタとサイト3の待ち時間の関係を示したものである。 $B_2 <= Bo_2$の場合、B_2の増加による通信のオーバヘッドの減少が大きく、W_2も減少する。$B_2 >= Bo_2$では、通信のオーバヘッドはなくなるが、B_2の増加により最初の待ち時間が増えるため、W_2は増加する。衛星通信のように、通信オーバヘッドが大きい場合、ブロッキングは有効である。

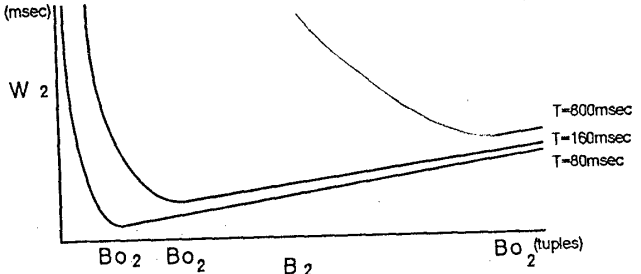

図3 ブロッキングファクタと待ち時間

4. おわりに

本稿では、パイプラインネストループ法におけるブロッキングファクタと処理効率について述べた。並列性を下げる要因に(1)並列動作に入るまでの待ち時間、(2)通信オーバヘッドがある。これらは、ブロッキングファクタに左右されることがわかった。(1),(2)は相反するものであり、ブロッキングファクタを増やすことで、(2)は削減できるが、(1)が大きくなる。さらに、考察を深め、最適なブロッキングファクタを得ることができた。

質問処理では、処理方法、処理プラン(結合の順番等)が重要である。従来、通信のオーバヘッドを考慮し、転送量の削減を目的としたプランの決定についての研究が多かった。今後、並列性を考慮したプランの決定も必要である。また、以下の2点を考慮した処理時間の見積りについても、検討していきたい。

①値のばらつきによる、結合のためのフェッチ回数のばらつき

②複数利用者による、1サイトへの負荷の集中によるフェッチ時間の変化

[1] Kiyoki, Y., Kato, K., Masuda, T., "A Stream-Oriented Approach to Distributed Query Processing in a Local Area Network", Proc. 1986 ACM SIGSMALL/PC Symp. Small Syst., pp. 146-155, May. 1986.

[2] Hikita, S., Kawakami, S., Haniuda, H., "A Stepwize Approach to Distributed Database Systems by Database Machines", Proc. 1985 ACM SIGSMALL Symp. Small Syst., pp. 18-24, Dec. 1986.

[3] Selinger, P. G., Astrahan, M. M., Chamberlin, D. D., Lorie, R. A., Price, T. G., "Access Path Selection in a Relational Database Manegement System", Proc. ACM SIGMOD 1979 Int. Conf. Management of Data, pp. 23-34, 1979