データベース・システム 64－6
（１９８８．３．１５）

分散データベースの質問処理最適化に関する考察

李　紅、　佐藤　洋

電気通信大学、情報工学科

分散データベースの質問処理について、優れた処理方法であり、応答時間について最適性を主張しているApers, Hevner & Yao (1983)のアルゴリズムでは、始めに関係を単一属性に射影して同一属性の結合についての効率のよいスケジュールを作り、これを各関係ごとに統合している。本研究では、このアルゴリズムの持つ欠点を３つの観点に立って指摘し、最適性についての反例を挙げている。その第一は、関係間の複数属性の結合が有利となること、第二は、一方の属性の結合と他属性の結合とが関係のサイズに影響を及ぼす効果、第三は、AHY のアルゴリズムで、ある関係から他の関係に複数の属性を分離して送るときの遅延を無視していることである。

A Note on Query Optimization of Distributed Database Systems

Hong LI, Hiroshi SATO

University of Electro-Communications

University of Electro-Communications,
1-5-1 Chofugaoka, Chofu-shi, Tokyo, Japan 182

Ingenious methods to optimize the response time and the total time for queries on a distributed database were proposed by Apers, Hevner and Yao (1983). Their algorithms are studied from three points of view and counterexamples to invalidate the optimality especially for the response time are shown. The first problem is an effect of a join between relations with respect to plural number of common attributes jointly. The second one is an effect of a join of one attribute on the join of another attribute. The third one is a time delay caused when different joining attributes of two relations are joined separately.

1．はじめに

コンピュータシステムやネットワークの技術の進歩につれて、分散データベースの問題が重要になってきている。分散データベースの理論については多くの重要な問題が残されている。分散データベースの質問処理の最適化はその中の一つである。この問題については従来多くの研究が行われている[1]～[7]。これらの研究は質問に対する強い制限（例えば、トリー質問に限るなど）を加えるもの[5][6]と比較的に一般的な質問に対して最適化を求めるもの[2]～[4]と質問処理の性質を見極めるもの[7]等がある。 本研究では一般的な質問の処理に関して先導的な役割を果たしているApers, Hevner, Yao(1983)（今後 AHYと略称する）の方法[3] について三つの角度から考察を行い、AHY[3]で提出されたアルゴリズム RESPONES が最適でないことをいくつかの例を挙げて説明する。

本研究の扱う分散データベースはデータが各属性に関して均一分布をとり、また属性が独立とする。質問処理を行う際、処理コストは通信コストに比べて無視できて、データの転送コストは転送するデータの量の線形型関数であると仮定する。ある質問に対して、回線使用時間はこの質問を処理する際、占有した通信回線の時間で、応答時間は質問処理の開始時間と処理の終わった時間の差である（待ち行列を考えないこととする）。通常、回線使用時間と応答時間は質問処理アルゴリズムを評価する二つの主な規準とされている。本研究ではこの時間を転送コストという。

本論文では第２節は記号と表示法、第３節は複数属性の結合について、第４節は一属性の結合と他の属性の結合との相互作用、第５節は AHYのアルゴリズムの第３の問題点について述べ、最後の節ではいくつかの議論について簡単に述べる。

2．記号と表記法

本研究では関係型データベースを扱い、R, R_a, R_1, R_2 等は関係を表すものとする。また、異なる関係は異なるノードにあるとする。 Rの持つ属性が aと bのとき、R(a,b)は Rと同じように関係を表すが、R(a)は Rの属性 aへの射影とする。属性 aと bの領域の大きさはD_a、D_bとし、属性 aと bの大きさ（バイト数）はそれぞれW_aとW_bとする。またt, t_a, t_b はそれぞれ R, R(a), R(b)のタプル数とし、 S, S_a, S_bは R, R(a), R(b)のサイズ（関係のデータ量）を表すこととする。従って、以下のような式が成り立つ。

$$S = (W_a + W_b) * t, \quad S_a = W_a * t_a, \quad S_b = W_b * t_b \tag{2.1}$$

そのほかに選択度u という概念があり、R の持つ属性が aと bである時、R(a,b), R(a), R(b)の選択度はそれぞれ u, u_a, u_bで表すが、u, u_a, u_b は下の式によって定義される。

$$u = t / (D_a * D_b), \quad u_a = t_a / D_a, \quad u_b = t_b / D_b \tag{2.2}$$

また、$R \xrightarrow{a,b} R_1$ は RからR(a,b)をR_1に送り、 $R \xrightarrow{a}$は Rから R(a) を送ることとなる。そのほかに、$R \xrightarrow{a}, \xrightarrow{b}$は RからR(a)とR(b)をシリアルに転送する意味とする。転送コストについては、転送コストC と送るデータの量 Sとは線形型関数であるが、便利のため

$$C = S \tag{2.3}$$

とする。→の周りには数字が付いている場合、転送コストを表す。従って、回線使用時間と応答時間は皆転送するデータの量で表せる。本研究では、質問の発生するノードを結果ノードと呼び、これを Tと書く。このノードは質問に必要な関係の存在するノードとは異なるものと仮定する。この仮定を緩めることに大きな困難はない。

3．複数属性の結合についての考察

3．1 複数属性を持つ関係の特性

関係が複数の属性を持つ時、それらの属性のサイズとその選択度はその関係の最も重要な特性である。ここでは属性 aと bを持つ関係 Rについて考えよう。式(2.2) より、R の選択度 u = t / (D_a * D_b)となる。通常、領域の積は大きい値をとるので、R の選択度は一般にかなり小さいと考えられる。R のタプル数 tについて考えると次のような不等式が成り立つ。

$$\max \{ t_a, t_b \} <= t <= t_a * t_b \tag{3.1}$$

従って

$$u <= u_a * u_b \tag{3.2}$$

となる。式 (3.1)を見て、関係R(a,b)のタプル数t が式 (3.1)の左辺の程度の時、R は属性について1次であると

言い、同様に tが (3.1)式の右辺の程度の時、R は属性について2次であるということにする。上記の1次と2次という性質の定義にはやや曖昧な点があり、定性なものに過ぎないが、関係の性質を表現するのに有用の概念と考えられる。このような性質は関係の持つ一貫制約に依存している。例えば、a と bの一方が関係のキーである場合は、関係が1次であり、また、R(a,b)が属性 aと bの間の関連を表すような場合は2次の関係となることが多い。ここで Rの選択度u が複数属性を一つの集合として統一的に取り扱うように定義されていることが重要である。以下、関係の1次と2次という概念を用いて複数の結合属性を持つ関係の結合の性質について考えることにする。

3．2　結合属性の集合と関係の次数を配慮した質問処理その一

AHY のアルゴリズムは、すべての関係を各単一属性に分解し、各単一属性の結合を最適に実行することを基礎としているが、本節3、2と次の3、3節ではいくつかの関係が複数の属性を共有するとき、この複数属性について結合を行うことが有利である場合があることを、実例を用いて示すことにする。

この節では二つの関係 $R_1(a,b)$と $R_2(a,b)$が異なるノードにあり、第3のノードでR_1とR_2の aと bについて結合した結果を求める場合に限って考えることにする。これからR_1は1次の関係で、R_2は2次の関係と仮定する。さらに、 $S_1 < S_2$ が成り立つものとする。この場合は一般に属性を分離させてから結合を行うより属性集合で結合するほうが有利である。R_1とR_2のタプル数、サイズと選択度の表示は次の通りとする。

関係	タプル数	サイズ	選択度
$R_1(a,b)$	t_1	S_1	u_1
$R_1(a)$	t_{1a}	S_{1a}	u_{1a}
$R_1(b)$	t_{1b}	S_{1b}	u_{1b}
$R_2(a,b)$	t_2	S_2	u_2
$R_2(a)$	t_{2a}	S_{2a}	u_{2a}
$R_2(b)$	t_{2b}	S_{2b}	u_{2b}

$t_1 < t_2$ のほかに $t_{1a} < t_{2a}$、$t_{1b} < t_{2b}$ も成り立つと仮定する。これからまず回線使用時間、次に応答時間について考察を行う。計算は表1の例題について行う。

	サイズ	タプル数	選択度	属性 a		属性 b	
	S	t	u	S_{ia}	u_{ia}	S_{ib}	u_{ib}
関係 R_1:	200	100	10^{-4}	100	0.1	100	0.1
関係 R_2:	$2*10^5$	10^5	0.1	1000	1	1000	1

$D_a = D_b = 1000$,　$W_a = W_b = 1$

表1　例題の分散データベース（その一）

回線使用時間

関係の次数を配慮すると、次の方法のみを考えればよい。

方法 1) R_1 [a b] R_2 [a b] T　　転送コスト $C_1 = S_1 + L$

方法 2) R_1 [a], [b] R_2 [a b] R_1 [a b] T　転送コスト $C_2 = S_{1a} + S_{1b} + S_2 u_{1a} u_{1b} + L$

方法 3) R_1 [a] R_2 [a b] R_1 [a b] T　　転送コスト $C_3 = S_{1a} + S_2 u_{1a} + L$

方法 4) R_1 [b] R_2 [a b] R_1 [a b] T　　転送コスト $C_4 = S_{1b} + S_2 u_{1b} + L$

（ $L = S_2 u_1 (= S_1 u_2)$、L はすべての結合を行った結果の転送コストである。）

R_1が1次関係である仮定により、S_1と $S_{1a} + S_{1b}$とは同程度である。

C_1とC_2を比べると$S_2 u_{1a} u_{1b}$分だけC_1のほうが小さい。また、 C_1 と C_3 を比べると、R_2が2次関係であるため、$S_2 u_{1a} > S_{1b}$ が成り立つことが多い。従って、C_1はC_3より小さい。C_4はC_3と対称であるため、同じ理由でC_1は通常C_4より小さい。この結果、関係が異なる次数を持つ場合に、1次の関係の複数属性を一緒に他の関係へ転送する方がより小さい転送コストが得られることとなる。以下、表1の分散データベースを用いて、その結果を示し、AHY(1983) の方法とも比べる。

表1の数字を代入すると、

$L = 2 * 10^5 * 10^{-4} = 20$

$C_1 = 200 + 20 = 220$

$C_2 = 100 + 100 + 2 * 10^5 * 10^{-1} * 10^{-1} + 20 = 2,220$

$C_3 = 100 + 2 * 10^5 * 10^{-1} + 20 = 20,120$

$C_4 = 100 + 2 * 10^5 * 10^{-1} + 20 = 20,120$

このように、C_1が圧倒的に小さいことに注目されたい。

AHY のアルゴリズム TOTALを用いると、次の結果となる。

属性a の転送方法

1: $R_1 \xrightarrow[100]{a}$ 100

2: $R_1 \xrightarrow[100]{a} R_2 \xrightarrow[100]{a}$ 200

属性b の転送方法

3: $R_1 \xrightarrow[100]{b}$ 100

4: $R_1 \xrightarrow[100]{b} R_2 \xrightarrow[100]{b}$ 200

関係R_1の転送方法

IFS: $R_1 \xrightarrow[200]{a\ b}$ 200

関係R_2の転送方法

$R_1 \xrightarrow[100]{a} R_2$

$R_1 \xrightarrow[100]{b} R_2$, $R_2 \xrightarrow[2000]{a\ b}$ 2,200

回線使用時間 = 2,200 + 200 = 2,400

本例題では AHYの方法で得られた結果はわれわれの結果より遙かに大きい。

応答時間

応答時間の最適化について、前述の方法 1)～4) のほかに次のような並列的に転送する方法を考えればよい。

方法 5) $R_1 \xrightarrow{a}, \xrightarrow{b} R_2 \xrightarrow{a\ b} T$

$R_1 \xrightarrow{a\ b} T$ 転送コスト $C_5 = S_{1a} + S_{1b} + S_2 u_{1a} u_{1b}$

方法 6) $R_1 \xrightarrow{a} R_2 \xrightarrow{a\ b} T$

$R_1 \xrightarrow{a\ b} T$ 転送コスト $C_6 = S_{1a} + S_2 u_{1a}$

方法 7) $R_1 \xrightarrow{b} R_2 \xrightarrow{a\ b} T$

$R_1 \xrightarrow{a\ b} T$ 転送コスト $C_7 = S_{1b} + S_2 u_{1b}$

方法 5), 6), 7) はそれぞれ方法 2), 3), 4)に似ており、また皆 2), 3), 4) より L分だけ小さくなっている。従って、方法 1), 5)～7)を比較すればよい。

1)と5)を比べると、$S_2 u_{1a} u_{1b} > S_2 u_1$ のため、C_1が小さい。1)と6)を比べると$S_{1b} << S_2 u_{1a}$, $L = S_2 u_1 < S_2 u_{1a}$ (R_1が１次の関係であるため) が一般に成り立つため、やはりC_1のほうが小さい。同様にC_1はC_7より小さいことになる。例題の分散データベースで計算するとともに AHY(1983)の提出したアルゴリズム RESPONSE とも比較することにする。

$C_5 = 100 + 100 + 2 * 10^5 * 10^{-1} * 10^{-1} = 2,200$

$C_6 = 100 + 2 * 10^5 * 10^{-1} = 20,100$

$C_7 = 100 + 2 * 10^5 * 10^{-1} = 20,100$

C_1 = 220は最適である。

AHY のアルゴリズム RESPONSE で計算してみると、次のような結果がえられる。

属性a の転送方法

1: $R_1 \xrightarrow[100]{a}$ 100

2: $R_1 \xrightarrow[100]{a} R_2 \xrightarrow[100]{a}$ 200

属性b の転送方法

3: $R_1 \xrightarrow[100]{b}$ 100

4: $R_1 \xrightarrow[100]{b} R_2 \xrightarrow[100]{b}$ 200

関係R_1の転送方法

IFS: $R_1 \xrightarrow[200]{a\ b}$ 200

関係R_2の転送方法

$R_1 \xrightarrow[100]{a} R_2$

$R_1 \xrightarrow[100]{b} R_2$, $R_2 \xrightarrow[2000]{a\ b}$ 2,200*

応答時間 = 2,200

AHY(1983) が [3]でRESPONSEが最適なアルゴリズムであると定理で述べたが。前述の例題が示すようにRESPONSEが最適とは言えないことが分かる。HY(1979)とAHY(1983) の考え方はすべての属性を始めに分離して考えるのである。即ち、各関係の同種な属性のデータについて最適な転送方法を求めてから、これらの各属性に対する転送法を使って、各関係のデータ転送スケジュールを組み立てる。HY(1979)は [2]で単純質問 (各関係は１つの結合属性だ

* R1からR2に並列に属性 aと bを送ることができない。第５節を参照すること。

け含んでいる）について最適な回線使用時間と応答時間のスケジュールを得た。AHY(1983) の方法はHY(1979)のに基づいたものである。AHY がすべての属性を分離することにより同一関係に属する属性の間にあるデータの対応関係が失われてしまうため、通信量が増大する。その上、AHY では $u_{1a}u_{1b}$ よりずっと小さい uで関係のサイズの減少を図ることが考えられていないし、属性集合を用いて結合あるいは準結合を行う方法も考えられていない。これはAHY のやりかたの第一番目の問題点にもなり、上の例では、RESPONSEの最適性に対する一つの反例となっている。１次関係のサイズは２次関係のサイズと一般かなり違う。それと同時に１次関係の選択度u は通常比較的に小さい。このような性質を持っている１次関係に対して、属性の集合あるいは全関係を一つの単位として結合をするべきである。逆に２次関係の全関係を送ることは普通大きいコストをもたらすことになるだろう。

３．３　結合属性の集合と関係の次数を配慮した質問処理その二

前節では、属性 aと bを持つ二つの関係について属性集合による結合が有利な場合もあることを示したが。属性 aと bは一つの属性と考えるなら AHYのアルゴリズムがよいではないかと言われる恐れがあるので、第三の関係R_0を付加してさらに少し考えてみよう。第三の関係R_0は属性 aだけ持ち、R_1とR_2と異なるノードにあるとする。すべての関係は aについて結合をし、R_1とR_2とは属性 bについても結合をする。その結果を Tに求めるものとする。R_0のサイズ、タプル数と選択度はS_0、t_0、u_{0a} とする。また、

$$u_{0a} < u_{1a} < u_{2a} \qquad (3.3)$$

が成り立つとする。このとき、回線使用時間でも応答時間でも考えられる方法は始めに $R_0(a)$を送ることである。ここでは、表２の例題の分散データベースを用いて応答時間と回線使用時間けについて考察を行うことにする。

	サイズ	タプル数	選択度	属性 a		属性 b	
	S	t	u	S_{ia}	u_{ia}	S_{ib}	u_{ib}
関係 R_0:	200	200	0.2	200	0.2	———	
関係 R_1:	10^3	500	$5*10^{-4}$	500	0.5	500	0.5
関係 R_2:	$2*10^5$	10^5	0.1	1000	1	1000	1

$D_a = D_b = 1000$,　$W_a = W_b = 1$

表２　例題の分散データベース（その二）

R_0をR_1に送ることは属性及び関係のサイズを最も小さくするので、まずR_0をR_1に送ることにする。そうすると、R_1はR_0と結合をした後にそのサイズ、タプル数等が以下のように変わってくる。

			属性 a		属性 b	
S	t	u	S_{ia}	u_{ia}	S_{ib}	u_{ib}
関係 R_1':	200	100	10^{-4}	100	0.1	100*	0.1

R_0をR_2に送るよりは $R_1(a)'$を送ったほうがよいので、R_0をR_2に送らないことにする。従って、R_1'を新しいR_1に見直すと、３、２節でのR_1とR_2に関する仮定を満たしていることになり、３、２節の例題と同じになる。従って、応答時間と回線使用時間のコストは皆C_1にR_0を送るコスト200 を加えばよい。即ち、

応答時間のコスト　＝　220 + 200 = 420

回線使用時間のコスト ＝ 220 + 200 = 420

これから AHYのアルゴリズム RESPONSE と TOTALを使って計算を行う。

RESPONSE

属性a の転送方法

1: $R_0 \xrightarrow[200]{a}$　200

2: $R_0 \xrightarrow[200]{a} R_1 \xrightarrow[100]{a}$　300

3: $R_0 \xrightarrow[200]{a} R2$

$R_0 \xrightarrow[200]{a} R_1 \xrightarrow[100]{a} R2$, $R2 \xrightarrow[100]{a}$　400

属性b の転送方法

4: $R_1 \xrightarrow[500]{b}$　500

5: $R_2 \xrightarrow[1000]{b}$　1000

---

* この数字の計算方法は第４節の (4.2)式、または[4] と [8]が参照できる。

関係R_0の転送方法

IFS: $R_0 \xrightarrow[200]{a} 200$

関係R_1の転送方法

$R_0 \xrightarrow[200]{a} R_1 \xrightarrow[200]{a\ b} 400$

関係R_2の転送方法

$R_0 \xrightarrow[200]{a} R_2$

$R_0 \xrightarrow[200]{a} R_1 \xrightarrow[100]{a} R_2$

$R_1 \xrightarrow[500]{b} R_2$, $R_2 \xrightarrow[10000]{a\ b} 10{,}500$

応答時間 = 10,500

TOTAL

属性a の転送方法

1: $R_0 \xrightarrow[200]{a} 200$

2: $R_0 \xrightarrow[200]{a} R_1 \xrightarrow[100]{a} 300$

3: $R_0 \xrightarrow[200]{a} R_1 \xrightarrow[100]{a} R2 \xrightarrow[100]{a} 400$

属性b の転送方法

4: $R_1 \xrightarrow[500]{b} 500$

5: $R_2 \xrightarrow[1000]{b} 1000$

関係R_2の転送方法

$R_0 \xrightarrow[200]{a} R_1 \xrightarrow[100]{a} R_2$

$R_1 \xrightarrow[500]{b} R_2$, $R_2 \xrightarrow[10000]{a\ b} 10{,}800$

関係R_0の転送方法

IFS: $R_0 \xrightarrow[200]{a} 200$

関係R_1の転送方法

$R_0 \xrightarrow[200]{a} R_1 \xrightarrow[200]{a\ b} 400$

回線使用時間 = 200 + 400 + 10,800 = 11,400

上の例で示しているように、関係の次数と言う考え方は一般の質問に対しても有効で、かつ違う次数の関係の結合に対してはかなり小さいコストをもたらす事ができる。即ち、関係の次数は一般性を持ち、有効な考え方と言えよう。また、AHY のアルゴリズムで得た大きなコスト自身は AHYの考え方に問題があると示しているだろう。

4．一属性での結合と他の属性の結合との相互作用

二つの関係の間で共通な属性について結合を行う時、普通その属性のサイズは小さくなるのは当然である。これらの関係の他の属性のサイズも小さくなる。この問題は的中問題（Hit ratio-problem ）[4][8]となる。例えば、$R_1(a)$ と$R_2(a,b)$ があり、$R_1(a)$ を$R_2(a,b)$のあるノードに送るとすると、結合が行われた後R_2のサイズ $S_2' = S_2 * u_{1a}$ （S_2は結合する前の R_2 のサイズ）、 $R_2(a)$のサイズ $S_{2a}' = S_{2a} * u_{1a}$ となり、R_{2b}のタプル数 t_{2b}' は次の式により決められる（Yao, 1977, [8]）。

$$t_{2b}' = t_{2b} * \left(1 - \prod_{i=1}^{t_2} (t_2(1-1/t_{2b})-i+1)/(t_2-i+1) \right) \qquad (4.1)$$

次の式は上の近似式としてよく使われる。

$$t_{2b}' = \begin{cases} t_2', & t_2' <= 1/2 * t_{2b} \\ 1/3 * (t_2' + t_{2b}), & 1/2 * t_{2b} < t_2' < 2 * t_{2b} \\ t_{2b}, & 2 * t_{2b} <= t_2' \end{cases} \qquad (4.2)$$

質問を処理する際、関係の結合に関するこの性質をうまく利用すれば、かなりよい結果が得られる。言い替えると、この性質は質問処理に対して無視のできないことである。次の例はこのことを示している。

	サイズ S	タプル数 t	属性 a S_{ia}	属性 a u_{ia}	属性 b S_{ib}	属性 b u_{ib}
関係 R_1:	400	200	200	0.2	100	1
関係 R_2:	500	500	500	0.5	————	
関係 R_3:	10	10	————		10	0.1

$W_a = W_b = 1$, $D_a = 1000$, $D_b = 100$

表３　例題の分散データベース（その三）

例：分散データベースは表３のものとする。R_1、R_2とR_3は異なるノードにあり、R_1とR_2は aについて結合をし、R_1とR_3は bについて結合をする。その結果は第４のノードで求められている。

まず、結合をするとき、非結合属性への影響を考えたやり方を示す。

$R_3 \xrightarrow{b} R_1 \xrightarrow{a\,b} T$　　$R_3 \xrightarrow{b} R_1$　のコスト = 10

$R_1 \xrightarrow{a\,b} T$　のコスト = 400 * 0.1 = 40

$R_1 \xrightarrow{a} R_2$　のコスト = 20　（式(4.2)より）

$R_2 \rightarrow T$　のコスト = 500 * (20 / D_a) = 10

応答時間 = 50 、　回線使用時間 = 80

上のスケジュールはまず R_3 を R_1 に送って結合が行われる。これから R_1 をノード Tに送り、これによって小さくなった $R_1(a)'$ を R_2 に送って R_2 と結合をした後ノード Tに送ることになる。次に AHYのアルゴリズムRESPONSEとTOTALを使って同じ質問に対して計算を行う。

RESPONSE

属性a の転送方法

1: $R_1 \xrightarrow[200]{a}$　200

2: $R_1 \xrightarrow[200]{a} R_2 \xrightarrow[100]{a}$　300

属性b の転送方法

3: $R_3 \xrightarrow[10]{b}$　10

4: $R_3 \xrightarrow[10]{b} R_2 \xrightarrow[10]{b}$　20

関係R_1の転送方法

$R_3 \xrightarrow[10]{b} R_1 \xrightarrow[40]{a\,b}$　50

関係R_2の転送方法

$R_3 \xrightarrow[10]{b} R_2$

$R_3 \xrightarrow[10]{b} R_1 \xrightarrow[10]{b} R_2$

$R_1 \xrightarrow[200]{a} R_2$, $R_2 \xrightarrow[100]{a\,b}$　300

関係R_3の転送方法

$R_3 \xrightarrow[10]{b}$　10

応答時間 = 300

TOTAL

属性a の転送方法

1: $R_1 \xrightarrow[200]{a}$　200

2: $R_1 \xrightarrow[200]{a} R_2 \xrightarrow[100]{a}$　300

属性b の転送方法

3: $R_3 \xrightarrow[10]{b}$　10

4: $R_3 \xrightarrow[10]{b} R_2 \xrightarrow[10]{b}$　20

関係R_1の転送方法

$R_3 \xrightarrow[10]{b} R_1 \xrightarrow[40]{a\,b}$　50

関係R_2の転送方法

$R_1 \xrightarrow[200]{a} R_2$, $R_2 \xrightarrow[100]{a\,b}$　300

関係R_3の転送方法

$R_3 \xrightarrow[10]{b}$　10

回線使用時間 = 50 + 300 = 350

上に示されているように応答時間ばかりではなく、回線使用時間についても AHYの結論に比べて大きい節約が行われたことが分かる。結合の非結合属性に対する効果はときに非常に大きいと言えよう。上の例では AHYの RESPONSE の最適性についてもう一つの角度から反例が示されたことになる。これは AHY の考え方が始めから全ての属性を分離させることによる属性間の相互作用を無視した結果によっている。この例では関係を三つ、属性を二つしか含んでいないのに、得られた結果は AHYのものより遙かに優れている。もしもっと多くの関係と属性を取入れば、AHY のアルゴリズムより、もっと低い転送コスト（応答時間と回線使用時間）が得られるに違いない。

なお、AHY のアルゴリズム RESPONSE に対しては単純質問の他、これをわずかに一般化した次のような限られた問題に対してはその最適性を証明することができる。

関係は $R_1(A,B_1)$, $R_2(A,B_2)$,...,$R_n(A,B_n)$であって、A は結合属性で、

B_i (i = 1, 2,...,n) は属性あるいは属性の集合で、次の条件を満たす。

$B_i \cap B_j = \Phi$　　i ≠ j 時、i, j = 1, 2,...,n

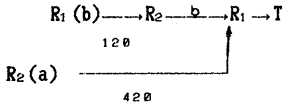
この問題は結合属性がただ一つであるため、前節で述べた複数属性での結合の可能性が存在せず、また本節で述べた他の属性についての結合との相互作用も存在しない。

5. AHY（1983）のアルゴリズムの第三の問題点

通常 POINT-TO-POINT 式のネットワークにおいては、ノードごとに IMP(Interface Message Processer) と通信

チャンネルが一つずつあり、一ノードが他ノードにメッセジを送る時、このノードがメッセジを自分の IMPに送り、IMP が送られたメッセジを記憶してから、他の IMPに送る。メッセジを受け取る IMPもメッセジを記憶し、他の IMP に転送する。このような過程をずっと結果ノードの IMPに到達するまで一回あるいは何回繰り返すことによってノード間の通信は行われる[1]。

しかし、AHY(1983) の論文[3] では RESPONSE について次のようなデータ転送スケジュールが存在している。



このスケジュールの意味は R_2(a)を R_1 に送る同時に R_1(b)を R_2 に送って、 R_2と準結合をしてから R_2(b) を R_1に送ることである。このスケジュールはR_2(a)とR_2(b) を並列的にR_1に送ることを含んでいる。でも、実際上にはR_2を含んでいるノードの IMPが R_2(a)と R_2(b)をR_1のあるノードの IMPにシリアル転送することとなる。AHY のアルゴリズムを使って、二つのノードの間に何重も並列的にデータを転送するスケジュールを作ることができるたろう。AHY のこのような考え方は実際の通信方式に反することは AHYのアルゴリズムの第三の問題点となる（第三節と第四節は AHYのアルゴリズムの第一と第二の問題点とも考えられる）。このような考え方に基づいているアルゴリズムと応答時間の計算は明らかに誤っていると思われる。

6．議論

分散データベースの質問処理問題はＮＰ困難であると予想されており、その回線使用時間と応答時間の最適な解を求めることはきわめて難しい。このような問題の解決へのアプローチとしては実際の通信方法をもとにして、複数属性による結合・準結合とその効果等を十分考えなければならない。ここでは、ある制限された場合に最適である応答時間についての AHYの方法に内在する三種の問題点を指摘した。この指摘は AHYの方法の改善の方向を示唆するものと言えよう。

また、応答時間と回線使用時間のそれぞれに対して、普遍的な最適解を求めることの困難を考え、質問に関与する関係について、関係の数や、属性の数等に制限を加えた問題の最適解を求めるようなアプローチも考えられよう。

7．参考文献

[1] S. Ceri and G. Pelagatti, "Distributed Database - Principles and Systems", McGraw-Hill Book Company, 1984.

[2] A. R. Hevner and S. B. Yao, "Query Processing in Distributed Database Systems", IEEE Trans. on Software Engineering, Vol. SE-5, No. 3, May 1979, pp. 177-187.

[3] P. M. G. Apers, A. R. Hevner, and S. B. Yao, "Optimization Algorithms for Distributed Queries", IEEE Trans. on Software Engineering, Vol. SE-9, No. 1, January 1983, pp. 57-68.

[4] P. A. Bernstein, N. Goodman, and E. Wong, et al., "Query Processing in a System for Distributed Databases(SDD-1)", ACM Trans. on Database Systems, Vol. 6, No. 4, December 1981, pp. 602-625.

[5] C. T. Yu, Z. M. Ozsoyoglu, and K. Lam, "Optimization of Distributed Queries", J. Computer and System Sciences 29, pp. 409-445, 1984.

[6] B. Gavish and A. Segev, "Set Query Optimization in Distributed Database Systems", ACM Trans. on Database Systems, Vol. 11, No. 3, September 1986, pp. 265-293.

[7] A. L. P. Chen and V. O. K. Li, "Improvement Algorithms for Semijoin Query Processing Programs in Distributed Database Systems",IEEE Trans. on Computers, Vol. C-33, No. 11, November 1984, pp. 959-967.

[8] S. B. Yao, "Approximating Block Accesses in Database Oragnizations", Communication of the ACM, Vol. 20, No. 4, April 1977, pp. 260-261.