

2H-8

データベースの並列分散処理 に関する一考察

井澤 忠 ・ 古賀 義亮
(防衛大学校)

1. まえがき

一部分の障害が発生したために、データベース全体にわたる被害を被る場合がある。この被害を局部的にとどめる方法として、データベースに冗長性をもたせて分割し、分散配置させる方法が考えられる。本研究では、リレーショナル・データベースについてその属性に冗長性をもたせて分割し、並列分散処理に適した3ポートネットワーク上に分散記憶させる方法について考察する。ここでは、3項分割により集合演算等を容易にし、フォールトトレラント化する一例を示す。

2. ネットワーク構造

本稿で取扱うコンピュータネットワークは、図1に示すように各ノードコンピュータが三本の入出力端子を持つ3ポートネットワークである。ノードコンピュータはそれぞれメモリを持ち独立した処理機能を持っている。三つの入出力ポートには専用の通信制御装置があり、ノードコンピュータ間のデータ伝送には対向の全二重回線通信方式を用いる。

このようなネットワーク型コンピュータにおいて、プロセッサ間の通信はすべて対向通信のため常にプロセッサ間の通信を確保できる。プロセッサ間の通信速度を適切に設定すれば、バス結合方式のように通信量に制限を加える必要はなく仕事の負荷をノードコンピュータに効率よく分散する¹⁾ことができる。

このネットワークは、次の条件を満足しているものとする。

- (1) 各局は他局の位置を知っている。
- (2) 各局において、データベースの保存及び集合演算が可能である。

ここで3ポートネットワークを取り上げた理由は、そのネットワークが最小次数の均一グラフで表わされ、しかも、ネットワーク上で2分木構造が容易に作成できるためである。

3 分割方式

(1) 多値従属と第4正規形

多値従属は次のように定義される。関係Rが属性 $A_1, \dots, A_m, B_1, \dots, B_m, C_1, \dots, C_m$ から成り立っているとし、それぞれ A, B, C で表わす。 $R[A_1, \dots, A_m]$ に含まれるタプル $\langle a_1, \dots, a_m \rangle$ を a で表わす。 B, C

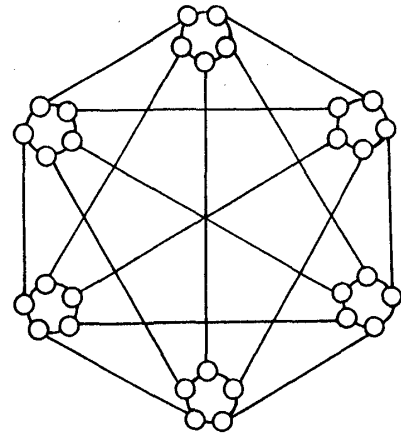


図 1 3ポートクラスタ型ネットワーク例

も同様とする。更に $\{b \mid \langle a, b, c \rangle \in R\}$ によって Bac を定義する。任意の a, c に対して $Bac = Bac$ であるとき、 B は A に多値従属であるという。言い換えると $\langle abc \rangle$ と $\langle ab'c' \rangle$ の二つのタプルが R 中に存在する時、必ず $\langle ab'c' \rangle$ と $\langle abc' \rangle$ も存在することを意味している。また、多値従属の無い形への分解を Fagin は第4正規形²⁾とよんだ。第4正規形に分解しても情報の損失はなく、従って結合演算によってもとの関係が再現できる。

(2) 3項分割方式

リレーションを細分化すればするほど障害が生じても局所化出来るが、管理が複雑になり通信制御のオーバーヘッドの問題が生じる。3項分割方式は、リレーション R (図2) の属性性を KAB, CDE, FKA の順に分割する方式である。

ここでは、ネットワーク上で属性の冗長度が2になる様に分割し、更にフォールトトレラントを目的としてリレーションが2項関係になるように分割する。ここで2項関係の属性値集合を2項ファイルと呼ぶことにする。この2項ファイルに、分割前のリレーションのキー属性を付加することにより3項分割にする。例ば、図2のリレーション R を分割すれば、 $KAB, KBC, KCD, KDE, KEF, KFA$ の3項関係が得られる。この3項関係の属性値集合を3項ファイルと呼ぶことにする。分解した3項ファイルは、キー属性に対し第4正規形をなし、結合演算を施すことによりもとのリレーションを再現できる。この1例は、

リレーションR

K	A	B	C	D	E	F
K1	A1	B1	X	D1	E1	F1
K1	A1	B1	Y	D1	E1	F1
K2	A2	X	C2	D2	E2	F2
K2	A2	Z	C2	D2	E2	F2
K3	A3	B3	C3	D3	E3	F3
K4	A4	B4	C4	X	E4	F4
K4	A4	B4	C4	Y	E4	F4
K4	A4	B4	C4	Z	E4	F4

K	A	B
K1	A1	B1
K2	A2	X
K2	A2	Z
K3	A3	B3
K4	A4	B4

K	B	C
K1	B1	X
K1	B1	Y
K2	X	C2
K2	Z	C2
K3	B3	C3
K4	B4	C4

K	C	D
K1	X	D1
K1	Y	D1
K2	C2	D2
K3	C3	D3
K4	C4	X
K4	C4	Y
K4	C4	Z

K	D	E
K1	D1	E1
K2	D2	E2
K3	D3	E3
K4	X	E4
K4	Y	E4
K4	Z	E4

K	E	F
K1	E1	F1
K2	E2	F2
K3	E3	F3
K4	E4	F4

K	F	A
K1	F1	A1
K2	F2	A2
K3	F3	A3
K4	F4	A4

図2 3項ファイルの一例

3項分割をする際にキー属性と他の2項の属性とで結合演算したものであり、キー属性によって検索速度を速めることのできる分割方式である。

4. 分散方式

分散方式には、木構造による方式とループ構造による方式等が考えられる。木構造分散方式では、処理を行う度に統制局を經由するためオーバーヘッドが増加し、統制局の障害対策が複雑となるばかりでなく、統制局の障害は致命的な被害をもたらす。ループ構造分散方式ではシステム管理は複雑になるが、障害に対し比較的強く3項関係をたどればループ構造となり、通信制御のオーバーヘッドが少ない等の利点がある。そこで、ループ構造を用いた分散方式の1例を以下に示す。

分割した3項ファイルをネットワーク上でループ構造を捜し出し、図3のネットワーク上に分散記憶させる方法²⁾を用いる。

ループ構造を捜し出す方法としては、次の様な方法が考えられる。リレーションがいくつの3項ファイルに分割できるかを調べ、その数以上の局数をもつループ構造を捜す。このため、ある局を根としてネットワーク上に木を生成する。同一局が何回出現してもよいが、根とした局が最初に出現するまでの経路が、最小局数のループ構造である。更に木の生成を行い、再び根とした局が出現するまでの経路が2番目に局数の少ないループ構造で

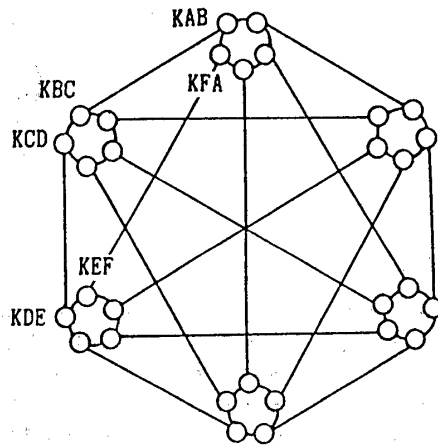


図3 分散配置例

ある。これを繰返し、分散するために必要な局数のループ構造を見出すことができる。

この方式では、ループを構成してリレーション及び3項ファイルを管理している各局は、3項ファイルがどの局で維持管理されているかを知っている。ループを構成しないネットワーク上の各局は、このループにリレーションが維持管理されていることを知っているのみで、どの局に3項ファイルが管理されているかは知らされていない。ループ外の局から3項ファイルの更新処理を要求する場合は、ループ上に一番近い局に依頼する。依頼されたループ上の局は、要求された3項ファイルを維持管理する局にたどりつくまでつぎつぎに依頼する。ここで、最初に依頼を受けたループ上の局においても処理することが可能である。

5. 障害発生時における処置

本分散方式では、3項ファイルをグループ管理しているので、ネットワーク内に障害が生じた場合を考える。

ある二局間で障害が発生した時、このループ構造は木構造に展開されるので、下位レベル局となる二つの局を見出せる。ここで、新たな木を生成し、最短経路を決定する。復旧した後、ファイル内部に障害があるならば、冗長度2で分散されているので、必要に応じて3項ファイルの修復を行う事ができる。

6. あとがき

ループ構造を用いて3ポートネットワーク上に分散させる方式では、システム管理が複雑になるが、障害に対し冗長度があるため比較的木構造に比べ強く、3項関係を辿ればループ構造となり、通信制御のオーバーヘッドが少ないなどの利点がある。

今後の課題として、複数リレーションの同時更新問題、分解方法並びにアクセスの最適化問題等について研究を進める必要がある。

参考文献

- 1) 竹之上、古賀；コンピュータネットワークによる並列論理シミュレータの一考察、情報処理学会(1984)
- 2) 有沢 博；データベース理論、(1980)
- 3) 古賀、山下、水上；リレーションナル・データベースの分散化に関する一考察、第8回FTC研究会