

問題適応型関係データベースマシンの アーキテクチャと性能評価

清水 康 田中若一 峰松彩子 磯田道男
小島清信 相磯秀夫 (慶應義塾大学工学部)

1. はじめに

1980年代に入り、コンピュータサイエンスの中でデータベースの占める位置はますます重要になってきており、この分野の研究はより実現性の高いものであることが要求されている。E.F. Coddによつて提案された関係モデルはデータベースモデルとして最も注目されている優秀なモデルであるが、その基本演算である関係演算を現行の汎用計算機で実現する場合、処理効率の悪さが問題となる。そこで関係演算の高速処理をねらった関係データベースマシン(RDBM)が数々提案されてきた。

初期のDBMはRAPI[OKKAWA]DBC[BANERJ]に代表されるlogic-per-track, logic-per-some track等のようにデータベース格納媒体(DBS: Data Base Store)に直接演算ロジックを付加した理想ディスク方式のものであった。これらのDBMは単純な関係演算(Selection, Restriction, Update)に対しては効果が大きいが、Join, Projection, Divisionのような複雑な関係演算に対しては処理効率が著しく劣下する。そこで、現在では複雑な関係演算を高速メモリ上でマルチプロセッサにより実行するRDBMが主流となってきた。[DEWI79][SCHU79]このような発展経過から現実的なRDBMの型態は次のようなものになると考えられる。

- 1) 単純な関係演算(Selection, Restriction, Column Selection(重複を排除しないProjection))のロジックはDBSに直接付加される。
- 2) 複雑な関係演算はDBSから分離した高速メモリ上でマルチプロセッサ(EMRO: Execution Module of Relational Operations)により処理される。
- 3) DBSとEMRO間のデータ供給を高速化するために両者間に中間バッファ

(SB: Staging Buffer)が配置される。

このような階層構成のRDBMにおける主要処理は、①DBS, SB間のデータ転送、②SB, EMRO間のデータ転送、③EMROにおける関係演算の並列処理、である。①についてはDBCで採用された可動ヘッドディスク1シリンドラ分の並列読み出し機構、さらにはDBCやSystem R [ASTR79]で用いられた転置ファイルによるDBSへの3クセス回数軽減等の有効な技術がある。また関係演算の並列処理方式に関係する②,③についてもいくつかの方式が提案されてきた。しかし、どの並列処理方式もすべての問合せに対して常に最適ではなく、その有効範囲はかなり限定されている。すなわち関係演算の並列処理方式として絶対的な支持を得るものは存在していない。過去のどのRDBMも、ある種の問合せに対し常に最適なものとなり得なかった原因は、各RDBMが常に1つの関係演算方式によつてしか問合せを処理できなかった点にある。こうした背景から、我々は様々な問合せに対して常に最適な関係演算方式を適用する問題適応型RDBMを提案する。本RDBMは関係演算方式として過去の多くのRDBMで採用された.Sort, Mergeによる方式、およびクラスリングによる方式[ODA80][KITSUB][KAWA81]を採用している。更に我々がすでに提案した独特な関係演算方式 V-method [KIYO79,81]をも提供している。本RDBMでは1タプルに1つのプロセッサを割り与えることにより関係演算を遂行することを基本としたVLSI指向の関係演算方式は採用していない。この方式は現行の技術をもつて実現するにはまだ困難があると考えられる。

以下、2.では各関係演算方式について述べる。3.では各方式の評価関数を定義する。4.では各方式の有効範囲を解析し、5.で

は、問題適応型RDBMの3-キテクチを示す。

2. 関係演算の並列処理方式

関係演算方式はSBからEMROの構成PE (Processor Element) 群へのリレーションの格納方法によって2つに大別できる。

1) 等分割配置方式: リレーションをPE台数に等しく分割し各部分リレーションを各PEへ1つずつ割り当てる。SBの構成ブロック数とPE台数が等しいならばSBとPE間でリレーションは並列転送される。

2) クラスタリング配置方式: リレーション内のタプルは演算対象属性のキー値に従ってPEに分割される。同じキー値をもつタプルは同じPEへ格納される。SBからPEへはクラスタリング・ロジック (ハッシュ回路 [ODA80], 多数の比較器によるタプル分類回路等) によつて、マクラスタリングされながら転送されるが、この転送をSBとPE群間の Inter Connection Network (ICN) によつて並列に行なうことを目指したものが発表されている。 [ODA80] [KITSU81]。しかしこの場合の並列度はデータ転送時の衝突やICNの結線方法によつて制限され、等分割配置方式の並列度よりは小さくなる。

本RDBMで採用した Sort-Merge による方式およびクラスタリングによる方式をJoinの処理例を用いて示す。また、V-method についてはJoinとProjectionから成る問合せ例を用いて代表的な演算を図示する。

2.1 分割ソート方式 (PSS: Partial Sort Scheme)

1) 2つの演算対象リレーション (Outer relation (O-relation), Inner relation (I-relation)) は各々等分割配置方式によりSBからPE群へ転送される。

2) 各PEは両部分リレーションのタプルを joining 属性についてソートする。

For $i=1$ to M do 3), 4), 5) (M : PEの台数)

3) PE# i 内の部分O-relationのソートされた全タプルが全PEへブロードキャストされる。

4) 各PEはPE# i からブロードキャストされた部分O-relationと、自PE内のソートされた部分I-relationをマージし、Joinの結合条件 ($=, <, >, \geq, <=, >=$) を満たすタプルを認識する。

5) 各PEは4)で条件を満たしたタプルを結合しJoin結果としての中間リレーション (T-relation) を生成する。

2.2 クラスタリング方式 (CLS: Clustering Scheme)

1) O-relation, I-relationの各タプルは、joining 属性のキー値に従いクラスタリング配置方式によつてSBからPE群へ転送される。

2) 各PEは自PE内の両部分リレーションを joining 属性についてソートする。

3) PE間の交信は一般に必要ないが、同じキー値をもつタプルが多く、1PE内にそれらのタプルを格納できない場合には同じキー値のタプルを格納するPE間での交信が行なわれる。

4) ソートされた部分O-relationと部分I-relationをマージしJoinの結合条件 (" $=$ "のみが可能) を満たすタプルを認識する。

5) 各PEは4)で条件を満たしたタプルを結合しT-relationを生成する。

6) T-relationは次に続く関係演算の演算対象属性についてクラスタリングするため、SBへ送り返される。この転送は並列度Mで行なうことができる。

2.3 V-method

V-methodはビットマップ表現された転置インデックスおよびリンク (2つのリレーションのタプルの結合関係を示す) により問合せを処理する独特な関係演算方式である。

V-methodは次のような特徴をもっている。

① リレーション内のタプルを一斉に識別するTIDの集合を表現するためにVTFが定義されている。VTFはリレーション内のタプルの物理的格納順序に従ってビットを並べたビット列である。N番目のタプルのTIDはNであり、VTF内のN番目のビットに対応する。

② 転置インテックス, リンクは複数の VTF を並べた形の VTFG, AIR によって各々表現される。

③ 複雑な関係演算に対応する Multiply-I, Multiply-II, Divide 演算が定義されている。Multiply-I は複数の リレーションにまたがっての Projection, Multiply-II は explicit-Join, Divide は Division に対応する。これらの演算により, explicit-Join を含む問合せの遂行においても中間リレーションを生成する必要がない。

④ これらの演算は 2 つの VTF 間の論理演算 (AND, または OR) を基本操作としてあり, 単純かつ規則的に処理できるので並列処理化が容易である。

V-method において扱うデータの構造を図.1 に, また例として VTFG の PE への分割配置状態を図.2 に示す。図.1 に示される UIL は演算対象の属性 (1 つまたは複数) に関して重複を排除した 3 アイテム値または複合アイテム値の並びである。UIL と VTFG の組は 1 つの転置インテックスに相当する。

V-method の演算には, Type 1: リレーションの属性値から VTF, UIL, VTFG, または AIR を生成する演算, Type 2: Multiply I, Multiply II, Divide のような 2 つの VTFG 間, または VTFG と AIR 間の演算, の 2 種類がある。Type I の演算はリレーションの演算対象属性を単分割配置方式により SB から PE 群へ転送し, ソートマージ方式により処理される。Type II の演算については, Join と Projection から成る問合せを Multiply II と Multiply I によって遂行する様子を図.3 に示す。実際には, VTFG, AIR は図.2 のように PE へ分割配置されており, Multiply I, Multiply II は並列処理により遂行される。この並列処理に際して VTFG, または AIR の全体像を認識するために図.2 で示した各 PE 内の CUIL 間で論理演算 (PE 間の交信) が必要となる。V-method の演算の詳細は, [KIY082] に示されている。

3. 関係演算方式の評価式

PSS, CLS については関係代数単位に, また V-method については各演算単位に実行時間を評価するための評価式を設定した。パラメータには, PE のハードウェア構成, SB と PE 間の ICN および PE 間の Inter Connection によって決まる物理パラメータと, 演算対象リレーションのサイズおよび内容によって決まるリレーションパラメータの 2 種がある。表 1 に物理パラメータ, 表 2 にリレーションパラメータを示す。また PSS, CLS については Join と Projection, V-method については Generate VTFG (VTFG の生成), Multiply I, Multiply II の評価式を表 3 に示す。これらの評価式では, 演算結果のリレーションは全 PE に等しく分割生成されると仮定している。また CLS においては, クラスタリングは

Relation Example

TID	NAME	CITY
1	Smith	London
2	Jones	Paris
3	Smith	Paris
4	Clark	London

RELATION R

TID	NAME	PART
1	Smith	Nut
2	Jones	Bolt
3	Jones	Nut

RELATION S

Data Structure

```

UIL (R.CITY)      :      { London, Paris } ~ UIL-1
VTF (R.CITY-London) :      TID 1 2 3 4
                   :      [ 1 0 0 1 ]
VTFG (R.CITY)    :      VTFG-1 (UIL-1)
                   :      [ 1 0 0 1 ] ~ London
                   :      [ 0 1 1 0 ] ~ Paris
AIR (R.NAME.S.NAME) :      R.TID 1 2 3 4      S.TID-1 2 3
                   :      [ 1 0 1 0 ] - Smith - [ 1 0 0 ]
                   :      [ 0 1 0 0 ] - Jones  - [ 0 1 1 ]
                   :      (UIL 3)
  
```

図.1 V-method のデータ構造

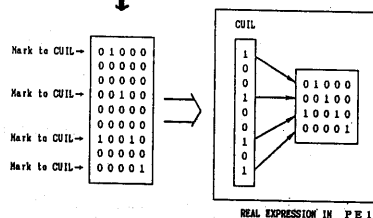
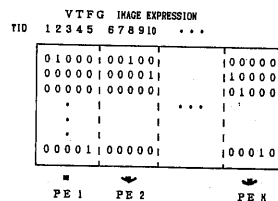


図.2 VTFG の PE への分割配置

Query Example

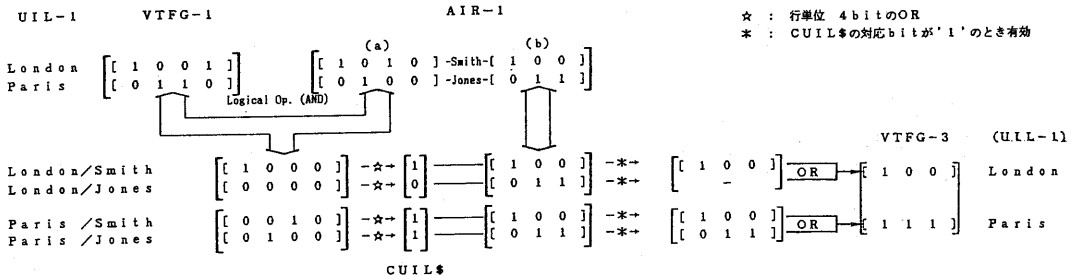
```
SELECT UNIQUE R.CITY, S.PART
FROM R, S
WHERE R.NAME = S.NAME
```

(REPRESENTATION OF RELATION ALGEBLA)
 Operation 1: Join [R.NAME=S.NAME]
 Operation 2: Projection [R.CITY, S.PART]

V-method

1. Generate VTFG (R.CITY) → VTFG-1, UIL-1
2. Generate AIR (R.NAME=S.NAME) → AIR-1, UIL-3
3. Generate VTFG (S.PART) → VTFG-2, UIL-2
4. MULTIPLY-II [(VTFG-1, UIL-1), AIR-1]
 → VTFG-3
5. MULTIPLY-I [(VTFG-3, UIL-1), (VTFG-2, UIL-2)]
 → RESULT OF THIS QUERY

MULTIPLY-II [(VTFG-1, UIL-1), AIR-1]



☆ : 行単位 4bitのOR
 * : CUIL#の対応bitが'1'のとき有効

MULTIPLY-I [(VTFG-3, UIL-1), (VTFG-2, UIL-2)]

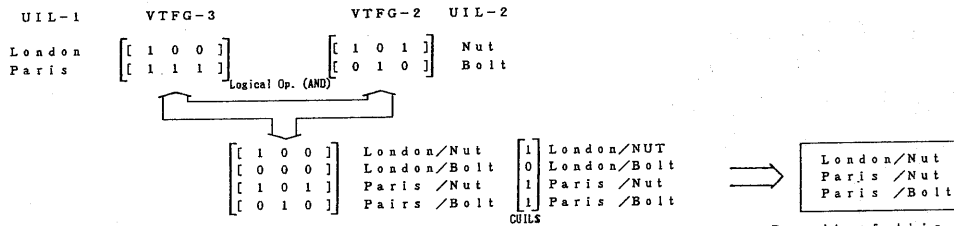


図.3 V-methodによる問合せ処理

表1 物理パラメータ

(1) Processing time parameters in PE.	
TR	: Time to fetch an item value in an internal memory of PE and to compare.
TW	: Time to fetch and write an item value in an internal memory of PE.
TSORT	: The coefficient of sorting time. Sorting time is expressed as $TSORT * n * \log(n)$.
Parameters only for V-method.	
TFR	: Time to fetch one bit in CUIL and to check "0" or "1".
TFW	: Time to write one bit in CUIL or VTFG.
TBR	: Average time per one bit to execute logical operation between VTFs.
TBW	: Average time to write one bit of VTF.
(2) Parameters for data communication time among PEs	
TBC	: Time to broadcast an item value from an internal memory of PE to every PE.
TFBC	: Average time per one bit to execute the logical operation among every CUIL in every PE.
(3) Parameters for data transfer time between SB and PEs.	
STM	: Time to transfer an item value from a block in SB to a corresponding PE. (equal-allocation method)
STD	: Time to transfer an item value in clustering to the destination PE from SB. (Clustering allocation method)
(4) Parallelism parameters	
M	: The number of PEs which can operate in parallel.
SH	: The degree of parallelism in clustering-allocation method.

表2 リレーション・パラメータ

(1) Parameters with respect to one relation	
N1, N2, N3	: The number of tuples in relation R1, R2, R3.
K1, K2, K3	: The number of distinct values on a joining attribute or the number of distinct tuples on projection attributes.
L1, L2, L3	: The number of attributes in relation R1, R2, R3.
LC	: The average number of attributes used in comparing for sorting or merging.
KK1, KK2, KK3	: The number of distinct values on a joining attribute of a partial relation in one PE or the number of distinct tuples on projection attributes of a partial relation in one PE.
Parameters only for V-method.	
KV1, KV2, KV3, KA1	: The number of distinct values in an operand attribute or the number of distinct compound-values in compound-attributes.
KKV1, KKV2, KKV3, KKA1	: The number of distinct values or distinct compound-values in one PE.
LV1	: The number of operand attributes in Generate VTFG.
q	: The ratio of the number of selected VTFs to the number of VTFs in AIR of Multiply-II.
(2) Parameters with respect to the relationship between two relations	
ALPHA	: Join selectivity factor. The number of tuples of a result relation by join operation is (N3). $N3 = N1 * N2 * ALPHA$

表3 関係演算並列処理方式の評価式

[1] Partial Sort Scheme [PSS]

PSS[Join]

1) transferring two operand relations from SB to PEs.

$$PSJSUP = STM*(L1*N1+L2*N2)/M$$

2) sorting in each PE on a joining attribute.

$$PSJSRT =$$

$$TSORT*\{(N1/M)*\log(N1/M)+(N2/M)*\log(N2/M)\}$$

3) broadcasting of tuples.

$$PSJBC = TBC*\min(L1*N1, N2*L2)$$

4) merging of tuples on a joining attribute of two relations.

$$PSJCMP = TR*(N1+N2)$$

5) generating result relation.

$$PSJTRG = TW*(L1+L2)*N1*N2*ALPHA/M$$

Total processing time :

$$PSSJOIN = PSJSUP+PSJSRT+PSJBC+PSJCMP+PSJTRG$$

PSS[Projection]

1) transferring an operand relation from SB to PEs.

$$PSPSUP = STM*L1*N1/M$$

2) sorting in each PE on projection attributes.

$$PSPSRT = TSORT*(LC*N1/M)*\log(N1/M)$$

3) broadcasting of tuples.

$$PSPBC = TBC*L1*K1$$

4) merging tuples on projection attributes.

$$PSPCMP = TR*LC*(K1+KK1*M)$$

5) generating result relation.

$$PSPTRG = TW*L1*K1$$

Total processing time :

$$PSSPROJECTION =$$

$$PSPSUP+PSPSRT+PSPBC+PSPCMP+PSPTRG$$

[2] Clustering Scheme [CLS]

CLS[Join]

1) Transferring two operand relations from SB to PEs.

$$CLJSUP = (STD/SM)*(L1*N1+L2*N2)$$

2) sorting in each PE on a joining attribute.

$$CLJSRT =$$

$$TSORT*\{(N1/M)*\log(N1/M)+(N2/M)*\log(N2/M)\}$$

The tuples in the operand relations are equally divided into every PE. This is an ideal environment for clustering scheme.

3) processing unnecessary.

4) merging tuples on a joining attribute of two relations.

$$CLJCMP = TR*(N1+N2)/M$$

5) generating result relation.

$$CLJTRG = TW*(L1+L2)*N1*N2*ALPHA/M$$

6) transferring the result relation from PEs to SB for executing the following operation.

$$CLJSDW = STM*(L1+L2)*N1*N2*ALPHA/M$$

Total processing time :

$$CLSJOIN = CLJSUP+CLJSRT+CLJCMP+CLJTRG+CLJSDW$$

CLS[Projection]

1) transferring an operand relation from SB to PEs

$$CLPSUP = (STD/SM)*L1*N1$$

2) sorting in each PE on projection attributes.

$$CLPSRT = TSORT*(LC*N1/M)*\log(N1/M)$$

3) processing unnecessary.

4) processing unnecessary.

5) processing unnecessary.

6) transferring the result relation from PEs to SB for executing the following clustering.

$$CLPSDW = STM*L1*K1/M$$

Total processing time :

$$CLSPROJECTION = SLPSUP+SLPSRT+SLPSDW$$

[3] V-method

V-method[Generate-VTFG]

1) transferring of one or more operand attributes from SB to PEs.

$$VGVSUP = STM*LV1*N1/M$$

2) sorting in each PE on the operand attributes.

$$VGVSRT = TSORT*(LC*N1/M)*\log(N1/M)$$

3) broadcasting of attribute values.

$$VGVCB = TBC*LV1*K1$$

4) merging attribute values and setting a bit on CUIL and VTFG.

$$VGVCMP = TR*(LC*K1+LC*N1)+TFW*K1+TFW*N1/M$$

Total processing time :

$$VGENVTFG = VGVSUP+VGVSRT+VGVCB+VGVCMP$$

V-method[Multiply-I]

1) checking bits of CUIL1 and CUIL2.

$$VMM1CC = TFR*(KV1+KKV1*KV2)$$

2) generating CUIL\$ by executing logical OR operation among CUIL\$'s of every PE.

$$VMM1BC = TFBC*KV1*KV2$$

3) executing logical AND operation between VTFs.

$$VMM1LOP = TBR*(N1/M)*KKV1*KKV2$$

4) comparing between CUIL\$' and CUIL\$.

$$VMM1CMP = TFR*KV1*KV2$$

5) Generating CUIL3 (corresponding to UIL3) and VTFG3.

$$VMM1WU = TFW*KV3+TBW*KKV3*(N1/M)$$

Total processing time :

$$VMMULTIPLY1 =$$

$$VMM1CC+VMM1BC+VMM1LOP+VMM1CMP+VMM1WU$$

V-method[Multiply-II]

1) checking bits of CUIL1 and CUIL2

$$VMM2CC = TFR*(KV1+KKV1*KA1)$$

2) generating CUIL\$ by executing logical OR operation among CUIL\$'s of every PE.

$$VMM2BC = TFBC*KA1*KV1$$

3) executing logical AND operations between two VTFs.

$$VMM2LOP = TBR*(N1/M)*KKA1*KKV1$$

4) comparing between CUIL\$' and CUIL\$.

$$VMM2CMP = TFR*KA1*KV1$$

5) generating CUIL3 (corresponding to UIL3) and executing logical OR operations between two VTFs (that is generation of VTFG-3).

$$VMM2WU = TFW*KV1+TBW*(N2/M)*KKA1*KV1*q$$

Total processing time :

$$VMMULTIPLY2 =$$

$$VMM2CC+VMM2BC+VMM2LOP+VMM2CMP+VMM2WU$$

理想的に行なわれ、各PEの処理時間が均等になっていると仮定している。しかし実際には、3アイテム値の分布表等を用いて各PEの負担が均等になるようにクラスタリングする場合でも理想的な処理時間の1.2~1.5倍の処理時間を要することをシミュレーションによって確認している。

4. 関係演算方式の評価

本章では、図3に示したJoin, Projectionから成る問合せ例の各関係演算方式による処理時間を比較検討する。この問合せにおいて、リレーションRとSをJoinした結果のリレーションをRSとし、このRSに対しProjectionが行なわれた結果のリレーションをRRとする。物理パラメータを次のように設定する。

(単位 μsec) $TR=1.0, TW=2.0, TSORT=3.0,$
 $TFR=0.01, TFW=0.01, TBR=0.01, TBW=1.0,$
 $TBC=2.0, TFBC=0.02, STM=3.0, STD=3.0.$
 $M=64$ 台, $SM=8$ (クラスタリング並列度)

リレーションパラメータは次のように設定する。R, S, RS, RRのタプル数は各々NR1, NR2, NR3, NR4とする。演算対象属性(1つまたは複数)内の区別できるアイテム値または複合アイテム値の種別数を各々KR1, KR2, KR3とする。

ここで $NR1 = NR2 = NO$ とし、KR1, KR2は $KR1 = KR2 = C1 \times NO$ ($C1 \leq 1$) とした。Joinの選択度 $ALPHA (= \frac{1}{C1 \times NO})$ は $1/NO \sim 100/NO$ まで変化させた。これはNR3をNOから100*NOまで変化させたことに等しい。またProjectionによる選択度は $NR4 (= KR3)$ がNO (R, Sのタプル数と等しい) になるように選んだ。Joinの結果のリレーションRSのタプル数NR3の変化に対する各関係演算方式の処理時間の推移を図4 (NO=16000) 図5 (NO=64000) に示す。

PSS, CLSの処理時間はNR3の増大に伴って増加する。反対にV-methodの処理時間は減少している。これはV-methodではリレーションRSを生成せず、さるには

処理されるAIR(R,S間のリンク)の大きさがALPHAの増加につれて小さくなるからである。V-methodは巨大な中間リレーションが生成される問合せにおいて有効であることがわかる。図4と図5の比較において、PSS, CLSの処理時間は $NO=16000$ と $NO=64000$ のときで約1:4であるのに対しV-methodでは約1:20になっている。

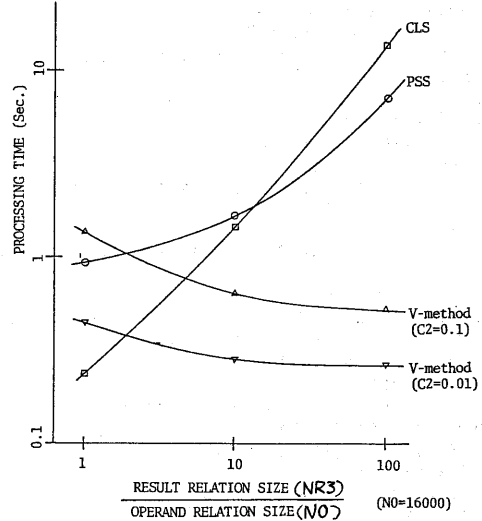


図4 問合せの処理時間

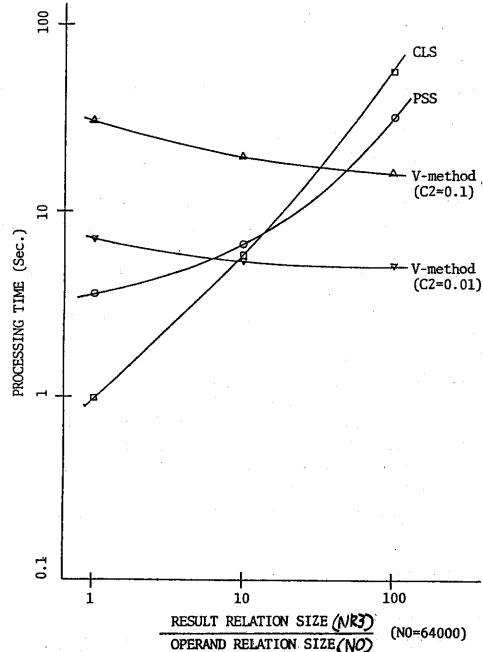


図5 問合せの処理時間

すなわち、PSSおよびCLSの処理時間はNOに比例し、V-methodの処理時間は $NO^2 \sim NO^3$ に比例している。従ってNOの増大にともないV-methodの有効範囲は狭くなっていく。PSSとCLSに関しては、NR3の増大にともないCLSの処理時間がPSSより急速に増大する。これは、PSSにおいて大きく処理時間を費やすPE間のブロードキャスト処理と、それに対応するCLSのSBからPEへのクラスタリング転送処理の性質の差による。PSSのブロードキャスト時間はJoinの場合はOuter-relationのタプル数(NO)に比例するがProjectionの場合には種別数(KR3: 今回一定)に比例する。一方CLSのクラスタリング転送時間はJoinの場合もProjectionの場合もタプル数(NO, およびNR3)に比例する。KR3 ≤ NR3であるので、NR3の増大につれてCLSの処理時間はPSSより急速に増大していく。

以上のように、最適な関係演算方式は問合せの性質や演算対象リレーションの内容によって異なってくる。

FP : Filter Processor
 I.B : Information Base
 PEi : Processing Element-i
 PEC : PE Controller
 SC : Staging Controller
 SBI : Staging Buffer Block-i

⇌ Data Line
 → Control Line
 ← Information

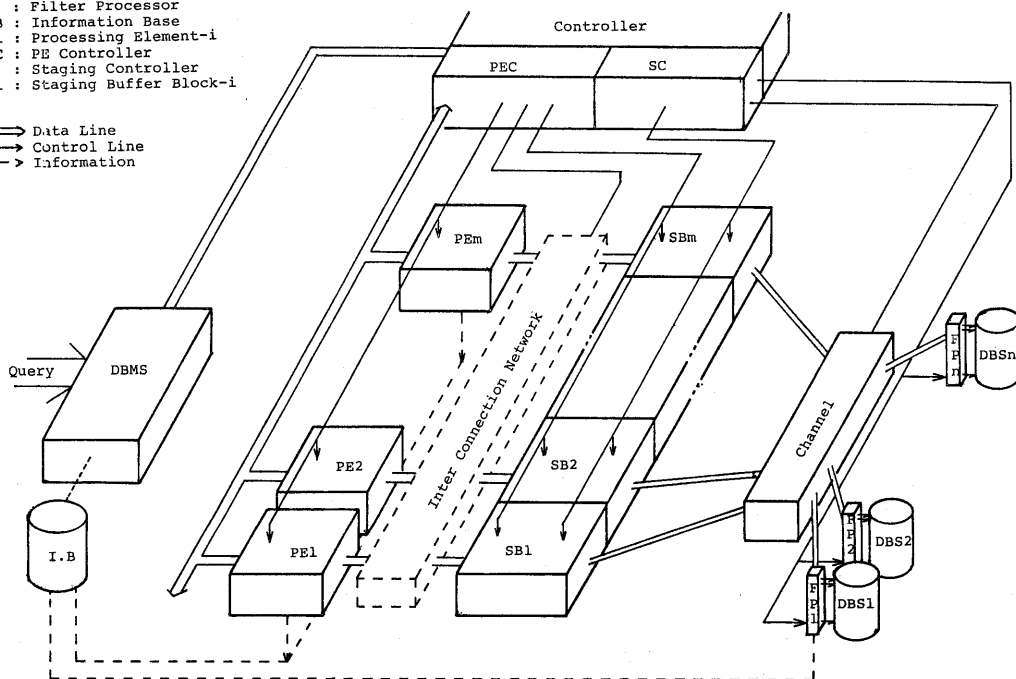


図5 問題適応型RDBMの3-キテクチャ

5. 問題適応型RDBMの3-キテクチャ

本稿で提案するRDBMの3-キテクチャの概観を図5に示す。このRDBMは、3つの関係演算方式PSS, CLS, V-methodを提供している。SB, PE間のInter Connection Networkは等分割配置とクラスタニング配置の両者が可能であるように構成される。3方式の中から最適な方式を選択するために、Data Base Mangement System (DBMS)は各関係演算に対して定義された評価式を用いる。これらの評価式のリレーションパラメータに代入される値はInformation Base (I.B)に格納される。関係演算の処理時間を前評価(pre-evaluation)することによりDBMSは最適な方式を選択し、PE Controller, および Staging Controller にその方式を指示する。この前評価プロセスは以前の関係演算の実行処理とオーバーラップできる。

6. おわりに

本稿では関係演算の並列処理方式の性能解析をもとに、ユーザからの問合せを最

適な方式により処理する問題適応型RDBMを提案した。この性能解析結果は、各関係演算方式の有効範囲が限定されており、すべての問合せに対し常に最適である方式は存在しないことを示した。これらことから、RDBM設計に際して次のような指針が得られた。

1) Data Baseへのアプリケーションが不変的、固定的であるならばRDBM設計者はそれらのアプリケーションに最も向く方式を選択すべきである。最適な方式を選択するために、本稿で示した関係演算方式の解析モデルを利用できる。

2) Data Baseに対するアプリケーションが変化に富み、様々であるならば本稿で提案したような問題適応性の概念を導入すべきである。

本稿で示したRDBMは各方式を同じPE上で実現するものであるが、プロセッサ、メモリ価格の減少にとともに各方式を各々専用のマルチプロセッサに担当させる機能分散型の形態をとるRDBMの実現性も高まるであろうと考えている。

謝辞

日頃、御指導いただいている慶應義塾大学工学部 所 真理雄博士に感謝いたします。また、常に貴重な御意見を述べてくださる広島大学工学部 上林襄行博士に感謝いたします。

[参考文献]

- [ASTR79] M.M.Astrahan, et al. : System R : A Relational Data Base Management System, IEEE COMPUTER, Vol.12, No.5, pp42-48, May 1979.
[BANER79] J.Banerjee, D.K.Hsiao, K.Kannan : DBC - A Database Computer for Very Large Databases, IEEE Trans. Comput., Vol.C-28, No.6, pp.414-429, June 1979.
[CODD70] E.F.Codd : A Relational Model for Large Shared Data Banks, Comm. ACM, Vol.13 No.6, pp377-387, June 1970.
[DEWI79] D.J.DeWitt : DIRECT - A Multiprocessor Organization for Supporting Relational Database Management Systems, IEEE Trans. Comput., Vol.C-28, No.6, pp.395-406, 1979.

- [DEWI81] D.J.DeWitt and P.B.Hawthorn : A Performance Evaluation of Database Machine Architectures, Proc. 7th VLDB, pp.199-213, 1981.
[KAMI79] N.Kamibayashi, H.Kato, Y.Kiyoki, H.Ozawa, K.Seo and H.Aiso : SPIRIT: A New Relational Database Employing Functional-Distributed Multi-microprocessor Configuration, 1st International Conference on Distributed Computing Systems, pp.759-771, 1979.
[KIYO81] Y.Kiyoki, K.Tanaka, N.Kamibayashi and H.Aiso : Design and Evaluation of a Relational Database Machine Employing Advanced Data Structures and Algorithms, Proc. 8th Symposium on Computer Architecture, pp.407-423, 1981.
[KIYO82] Y.Kiyoki, M.Isoda, K.Kojima, K.Tanaka, A.Minematsu and H.Aiso : Performance Analysis of Parallel Processing Schemes for Relational Operations and Adaptive Database Machine Architecture, Technical Report (Keio Univ.), 1982
[OZKA75] E.A.Ozkarahan, S.A.Schuster, K.C.Smith : RAP-An Associative Processor for Data Base Management, Proc. AFIPS NCC Vol.44, 1975
[SCHU79] S.A. Schuster., et al. : RAP.2 - An Associative Processor for Databases and Its Applications, IEEE Trans. Comput., vol. c-28, No.6, pp.446-457, 1979
[SEO81] K.Seo, N.Kamibayashi, A.Minematsu and H.Aiso : A Look-ahead Data Staging Architecture for Relational Database Machines, Proc. 8th International Symposium on Computer Architecture, pp.389-406, 1981.
[SU75] S.Y.W.Su, G.L.Lipovski : CASSM: a Cellular System for Very Large Databases, Proc. Int. Conf. on VLDB, 1975
[TANA80] Y.Tanaka., et al. : Pipeline Searching and Sorting Modules as Components of a Data Flow Database Computer, Proc. IFIP-80, pp.427-432, 1980
[ODA80] 小田 : データ分配ネットワークを用いたデータベース計算機の構成法, 信学技報 EC 80-12, 1980
[KAMI81] 上林, 瀬尾 : 記憶階層に關係代數演算の前処理機構を組み込んだ關係データベースマシンの基本アーキテクチャ, 信学技報 EC-81-49, 1981
[KITSU81] 喜連川, 鈴木, 田中, 元岡 : HashとSortによる關係代數マシン, 信学技報 EC-81-35, 1981
[KIYO79] 清木, 田中, 上林, 相磯 : 動的マクビットによるリレーショナル演算の並列処理方式とその評価, 信学技報 EC-79-66, 1979