3. アーキテクチャおよびハードウェアシステム ——並列推論マシンと知識ベースマシン

ICOT研究所第1研究室長　村上国男
ICOT研究所主任研究員　角田健男
ICOT研究所主任研究員　尾内理紀夫

アブストラクト　並列推論マシンと知識ベースマシンは，第五世代コンピュータシステムの中核のハードウェア要素である．ICOTで進められている第五世代コンピュータシステムに関する10年間の研究のうち，前期3年間は，システム構成する個々の要素に関する基礎研究の期間である．

本論文では，前期における並列推論マシンと知識ベースマシンに関する研究・開発の現状と成果について述べる．

1. はじめに

第五世代コンピュータシステムに関する研究・開発の目標は，知識ベースに基づく問題解決や推論を効率的に実行する事のできる知識情報処理システムのプロトタイプを実現する事である．

この為，10年間の研究・開発期間が設定され，この10年が3期の研究ステップに分割された．

このうち，前期3年間の目標は，システムを構成する各々の要素に対する基礎研究を遂行し，中期に実現するサブシステムの為の基本的な構成技術を確立する事である．

並列推論マシンと知識ベースマシンは第五世代コンピュータシステムの最も重要なハードウェア構成要素である．

最終的なシステムでは，この2つのマシン機能は相互に密に結合，統合されると予想される．

しかし，前期では，個々の要素技術の研究・開発が主であり，中期で構築する推論サブシステム，知識ベース・サブシステムのハードウェアに関する基本技術を確立する事が目的である．よって，並列推論マシンと知識ベースマシンの研究・開発テーマを独立に設定し，検討を進めている．

前期において，並列推論マシンの研究開発は，下期の三つの評価検討項目について行うことを目標とした．

（1）　推論操作の並列実行を制御する並列型推論基本メカニズム

（2）　データフロー概念に基づき推論操作を高速実行するデータフローメカニズム

（3）　抽象データ型の概念に基づく種々の言語機能を支援する抽象データ型メカニズム

前期3年間における並列推論マシンの研究開発状況は，概略以下の通りである．

（1）　並列型推論基本メカニズムの研究開発は，基本メカニズムとして，リダクション，節単位処理，完全コピーを選択し，メカニズムの基本検討と並列推論マシンの動作環境の解析，およびソフトウェアシミュレーションによる評価を行い，現在，シミュレーション専用装置の試作を行っている．中期前半において，この試作機により各種データを収集し，デ

ータフローメカニズムと合せて比較評価を行う予定である．

（2） データフローメカニズムに関しては，基本検討そしてソフトウェアシミュレーションによる評価を行い，現在，実験機を試作している．中期前半において，この試作機により各種データを収集し，データフローメカニズムを評価する．

（3） 抽象データ型メカニズムに関しては，核言語第1版の中で，概念の整理と明確化の検討を行った．

第2章では，並列推論マシンに関する研究成果，特にプログラムの特性解析結果とリダクション方式およびデータフロー方式のアーキテクチャの概要を述べる．

前期において，知識ベースマシンの研究開発は下記3つの評価検討項目について行うことを目標とした．

（1） 知識ベース演算の実行の全体的管理を行う知識ベースメカニズム

（2） 迅速な知識格納，検索，更新を行う並列型関係・知識演算メカニズム

（3） 大容量の知識格納および管理を行う関係データベースメカニズム

前期3年間における知識ベースマシンの研究開発状況は，概略以下の通りである．

（1） 知識ベースメカニズムの研究開発は，知識ベース演算に関する知見を充分に把握した後に研究を進めるべきであると判断し，前期に開発した関係データベースマシンDeltaを使用した各種実験結果に基づき中期初頭より開始させる予定である．

（2） 並列型関係・知識演算メカニズムの研究開発のうち，並列型関係メカニズムについては，複数台の逐次型推論マシン（PSI）とDeltaとを比較的粗なLANインタフェースを介して接続し，Delta内に最大4台の関係データベース・エンジンを並列に動作させる実験環境を確立させた．さらに，PSIとDelta間をLANを介しない密なインタフェースで接続動作させる実験環境を実現中であり，DeltaとPSI間密結合実験によるデータ収集および知識演算メカニズムの研究は，中期前半に展開させる予定である．

（3） 関係データベースメカニズムについては，知識ベースマシンに向けての基礎的技術を確立させるための各種実験を行うためと，大容量データベースをサポートするDeltaとPSIとをLANで結合したICOTソフトウェア開発システムを構築するという，2つの目的から関係データベースマシンDeltaの開発を行った．

第3章では，知識ベースマシンに関する研究成果，特に前期で開発した関係データベースマシンのアーキテクチャの概要を述べる．

2. 並列推論マシン

本研究開発は，第五世代コンピュータシステムの中核となる推論処理を並列に実行するマシンの実現をその最終目標（後期目標）とする．この最終目標に至るアプローチの一つの大きなマイルストーンとして中期並列推論マシン*がある．

前期においては，この中期並列推論マシンの実現のための基礎研究開発を行う．

2.1 前期研究課題

前期研究の目標は，並列推論マシンが動作する環境を解析し，マシンの設計条件を明確化すること，および，マシン実現のための構成方式を明確にすることである．

（1） 核言語の特性解析

核言語のベース言語であるProlog，Concurrent Prolog［Shapiro 83］で記述されたプログラムの静的，動的特性の解析を行う．そして，この結果を並列推論マシン・アーキテクチャの設計条件に反映させる．

（2） 各種並列推論方式の検討

各種並列推論方式の検討と，それら方式に基づく並列推論マシンアーキテクチャの検討，ソフトウェアシミュレーションによる有効性の評価，モジュール数8～16の実験機の試作による評価，および構成技術の検討を行う．

2.2 核言語特性解析

核言語のベース言語の一つであるPrologで記述されたプログラムの静的，動的解析を行い各種データを収集した．［Onai 84-1］

2.2.1 静的解析

（1） 収集データ

プログラムを先頭から読み込み，以下に列挙した各種データを収集する．

＊ OR relation 数（同一head述語シンボルを持ち，引

* 並列推論のための実行制御機構を備えた，100台規模のモジュールからなる一部LSI化された並列推論マシン

数個数が同一の clause 同志を OR relation にあるという)

* AND リテラル数
* head 述語の引数個数と，そのうちの構造体データ引数個数
* body 側の引数個数と，そのうちの構造体データ引数個数
* 組込み述語使用頻度
* Cut 数

また，clause を次の二種類に分けて解析を行った．

inference clause：OR relation にある clause の中に少なくとも一つの rule を含む clause

database clause：OR relation にある clause がすべて unit clause のみから構成される clause

(2) 静的解析結果

(a) inference clause の静的解析結果

ICOT で開発された 33 本の inference clause 主体の Prolog プログラム (ex. 形態素解析プログラム，述語論理式簡単化プログラム，BUP トランスレータ等) を静的に解析した．結果は次のとおりである．

* 1 clause 当たりの平均 cut 数 0.65
* 平均 OR relation 数 2.7
* 平均参照数 3
* head の平均引数個数 3
* head 述語引数の内の構造体データ比 0.2(20 %)
* 平均組込み述語数比 0.5(50 %)
  (unification を必要としない組込み述語数比 0.25)

(b) database clause の静的解析結果

ICOT で開発された 6 本の database からなる Prolog プログラム (ex. 辞書プログラム等) を解析した．主な結果は次のとおりである．

* 平均 OR relation 数 10
  (inference clause の場合の約 4 倍)
* head の平均引数個数 2.8

2.2.2 動的解析

まず，逐次型 Prolog (DEC-10 Prolog) プログラムを並列実行可能なように書き変える．この並列実行可能な Prolog プログラムにゴールを与えて実行させ，その過程で静的解析時のデータ (除 cut 数) と OR 並列度とを収集した．

ICOT で開発された 2 本(形態素解析プログラム，論理式の簡単化プログラム) のプログラムを並列 Prolog に書き換えて，動的に解析した．結果を表 1 に示す．

表 1

	形態素解析プログラム	論理式の簡単化プログラム
動的 OR relation 数	7.1	4.6
構造体データ比(%)	32	50
組込み述語数比	0.4	0.8
OR 並列度	8.3	3.2

2.2.3 結果のまとめ

(1) DEC-10 Prolog は，プログラム実行時に使用可能なメモリ空間も十分でないので，cut が多用されており，決定的なプログラムとなっている．しかし，assert/retract が本質的に使われていない場合などには，これを並列実行可能なプログラムに書き換えることができ，ある程度の並列性を得ることができる．

(2) AND リテラルの約半分は組込み述語であり，組込み述語の実行速度がプログラム実行速度に影響を及ぼす．

(3) database clause の OR relation 数は inference clause のそれの約 4 倍である．よって，大規模 database clause を含むプログラムの場合，database clause に関する unification の高速化が重要である．

2.3 マシン・アーキテクチャ

現在，4 つの方式について検討を進めている．[ICOT 84]

① リダクション方式 [Onai 84b]

核言語のベース言語である Prolog あるいは Concurrent Prolog プログラムの実行過程は goal と clause からの resolvent の生成の過程であり，これは goal が，clause というルールを用いて自らを modify することとみなせる．一方，リダクション過程とは，self-modification である．このように，Prolog あるいは Concurrent Prolog プログラムの実行過程とリダクション過程との間に親和性の良さを見い出すことができる．そこでマシン・アーキテクチャとしてリダクション方式を採用した．リダクション方式は，Prolog プログラムを OR 並列に，Concurrent Prolog を AND 並列に実行する．

② データフロー方式 [Ito 83], [Ito 84]

処理に必要なデータがすべてそろえば，それが引き金となって実行が開始されるというデータフロー概念は，プログラムに内在する並列性を，プログラマが陽に指定しなくとも引き出すことを可能とする．本方式は，この概念に基づき，核言語プログラムを並列に実行する．

③ 完全コピー方式

リダクション方式の一つと見なすことができる．リダクション可能なサブゴールを含むプロセス全体をコピーし，ユニットへ転送する．このため，コピー量の増加，ネットワーク上のパケット長の増加を招くが，個々のプロセスの独立性が高くなる．

④ 節単位処理方式

暇な処理ユニットがビジーな処理ユニットへ処理を要求する．ビジーな処理ユニットはそれに応じて，処理を切り出し，暇な処理ユニットへ送る．よってこの方式では，資源要求の爆発を回避することができる．しかし，全処理ユニットがビジーになるまでにある程度の時間がかかる．

2.3.1 リダクション方式並列推論マシン (PIM-R)

(1) 全体構成

図 2.1 に示すように Inference Module と，Structure Memory Module そしてそれらを連結するネットワークからなる．

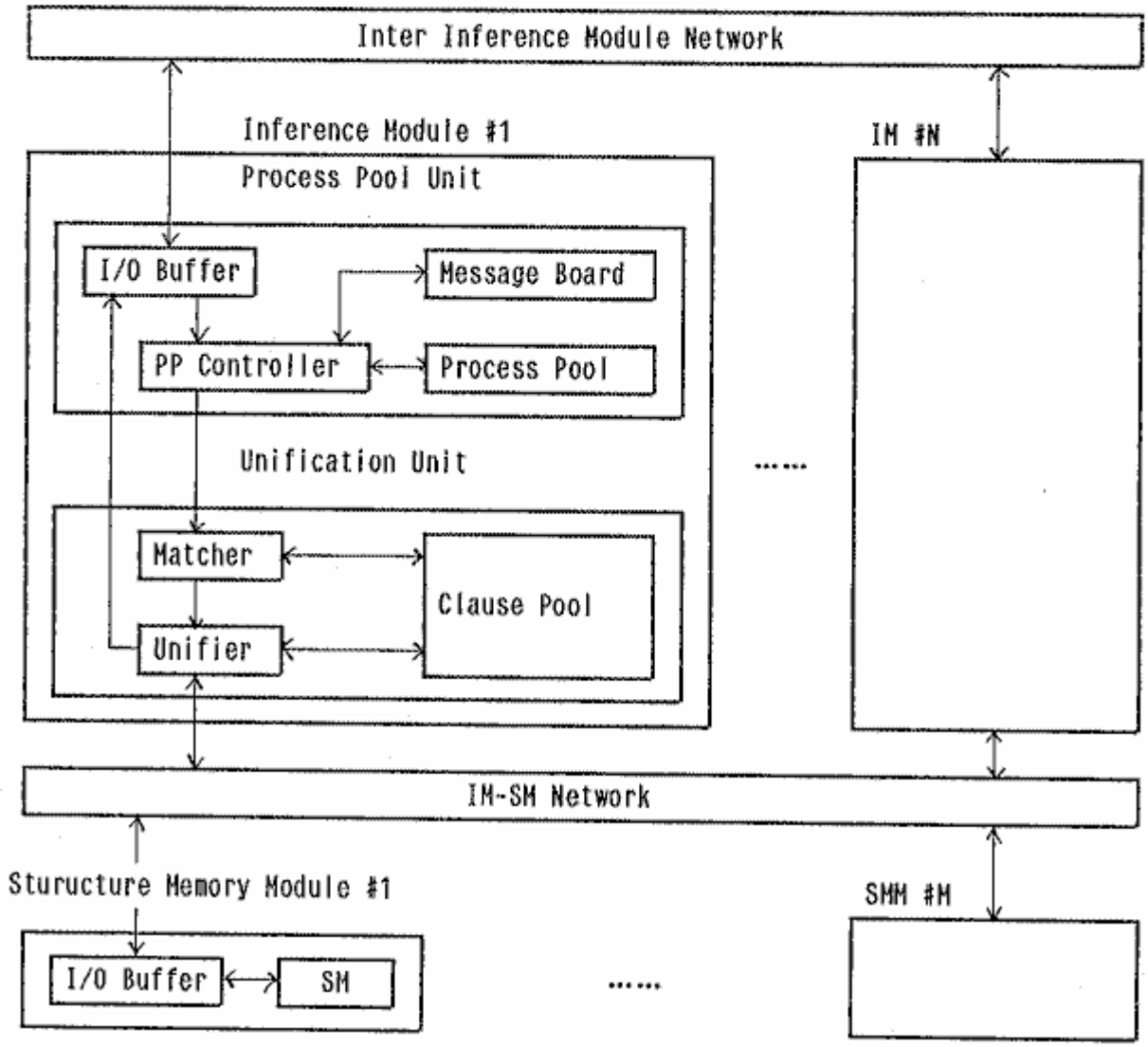


図 2.1 リダクション方式 PIM-R の概念構成

(a) Inference Module

この Module は，Process Pool Unit と Unification Unit とからなる．

＊ Process Pool Unit

Process Pool には，プロセス間の関係が格納される．リダクション可能なプロセスは Controller により選択され，Unification Unit へ送られる．リダクション可能かどうかは，sequential AND オペレータ，parallel AND オペレータ，commit オペレータ，producer プロセスからのメッセージ等によって指示される．

Message Board は，プロセス間のチャネルを通じて送られる値と，サスペンドプロセスを格納する．consumer プロセスがサスペンド状態になった時は，この Message Board に producer 側から値が送られて来ているかどうかをチェックする．未だ値が送られてきていなければサスペンドプロセスリストに自らをつなぐ．一方，producer プロセスがチャネルに値を結合すると，Message Board 該当セルにその値が書き込まれ，値を待っているサスペンドプロセスがあればそれに値を送る．

＊ Unification Unit

本ユニットでは，Process Pool Unit から送られてきたゴールの述語名と引数の個数および第一引数のデータタイプにより，Matcher において，あらかじめ unifiable clause の絞り込みを行う．その後，Unifier においてゴールと clause との unification を行い，結果を Process Pool Unit へ返す．clause および clause に関する情報は，Clause Pool に格納される．

(b) Structure Memory Module

本ユニットは構造体データを格納し，unification に必要な構造体データを Unification Unit へ転送する．

(2) アーキテクチャ上の特徴

＊ Process Pool 内のプロセスは 1 つ以上のサブゴール(body リテラル)から構成されるが，本方式では，リダクション可能なサブゴールのみをコピーして Unification Unit へ送る方式(部分コピー方式)を採用している．その結果，プロセス全体をコピーする方式に比べ，コピー量が少なくなり，またネットワーク上のパケット長も短くなる．

＊核言語第一版(KL1)のベース言語である Prolog を OR 並列に，Concurrent Prolog を AND 並列に実行する機能を持ち，KL1 を本マシン上で統一的かつ効率的に処理できる．

2.3.2 データフロー方式並列推論マシン (PIM-D)

(1) 全体構成

図 2.2 に示すように，マシンは推論の基本処理を行うための Processing Element Module(PEM) 群，構造データの格納や管理を行うための Structure Memory Module(SMM)群，及び，これらを結ぶネットワークから構成される．

(a) Processing Element Module

命令の実行制御，手続きの呼出し制御，基本パターン

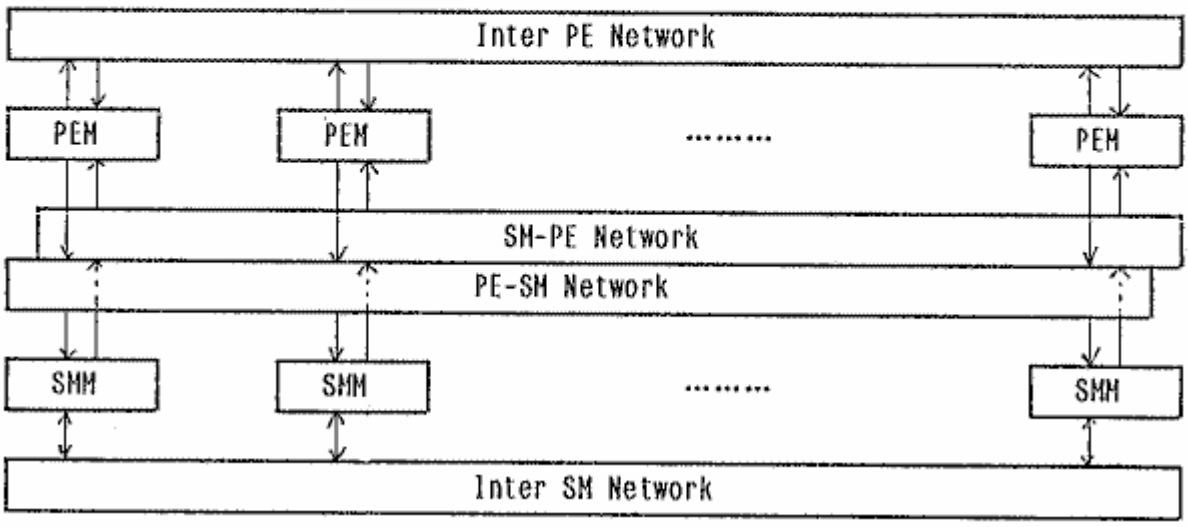


図 2.2 データフロー方式の概念構成

マッチング制御，および組込み述語実行の機能を有する．以下に述べる二つの部分からなる．

＊ Instruction Control Unit(ICU)

ICU は結果パケット（トークン）が到着した時に起動される．その結果パケットは，そのパケットの属するプロセス識別子，結果の送り先の命令のアドレスを示すあて先，および，結果の値から構成されている．ICU は，この結果パケットをもとにして，あて先で指定された命令の全オペランドがそろったか否か，すなわち，命令が実行可能となったか否かの判定を行う．命令が実行可能であれば，命令パケットを生成し，次に述べる Execution Unit へ送る．

＊ Execution Unit(EXU)

ICU から送られてくる命令パケットを受けとり，命令を解釈，実行し，結果をパケット化し，次のあて先へ転送する．

（b） Structure Memory Module

SMM は，ネットワークを介して EXU からの構造メモリ命令パケットが到着した時に起動され，参照カウント制御命令，データの読み書き命令，フリーセル取出し命令を実行する．

（2） アーキテクチャ上の特徴

本マシンの第一の特徴は，論理型プログラムに内在する 3 つのタイプの並列性(OR 並列性，AND 並列性，及び，統一化処理における並列性）をマシン上で実現していることである．

本マシンのもう一つの特徴は，統一化処理を行うプロセス間で構造データを共有する方式を採用していることにある．即ち，構造データは SMM 群に分散して格納され，プロセスを実行する PEM 群にはそのアドレス(ポインタ)が渡される．PEM は構造データアクセスの必要が生じた際に(オンデマンドで)SMM をアクセスする．これによって，複雑な構造データの処理を必要とするような応用における構造データコピーのためのオーバヘッドの問題が回避できる．

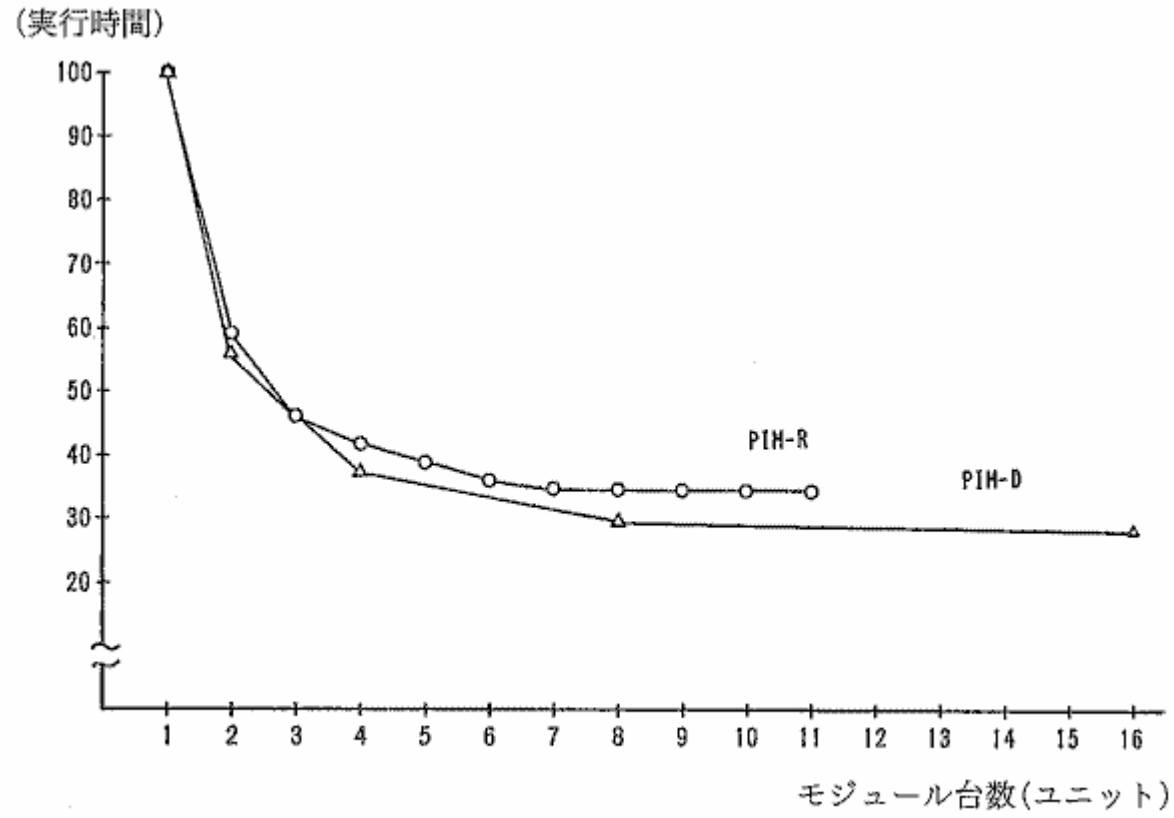


図2.3 4 Queens プログラムのシミュレーション結果（台数効果）

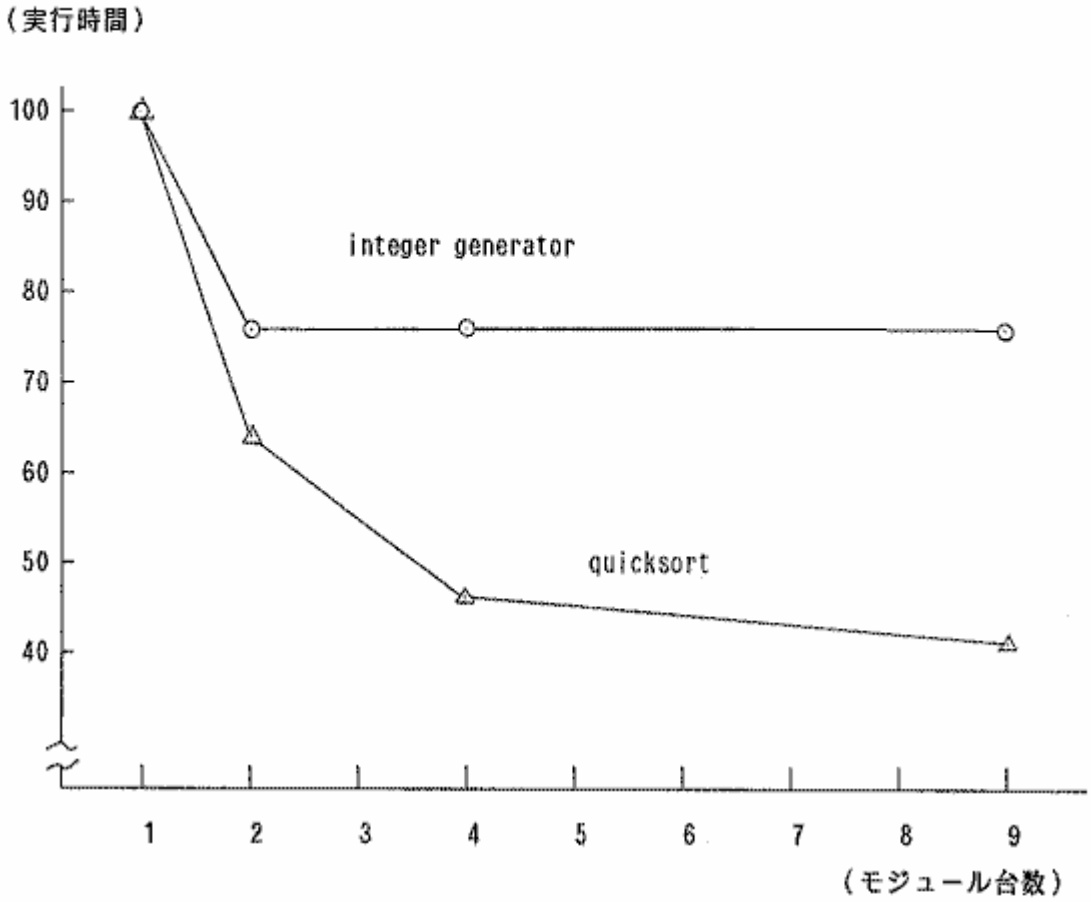


図2.4 リダクション方式 PIM-R による Concurrent Prolog プログラム・シミュレーション

2.3.3 ソフトウェアシミュレーションによる評価例

データフロー方式(PIM-D)，リダクション方式(PIM-R) に関してそれぞれソフトウェアシミュレータを開発し，シミュレーションを行った．(データフロー方式ソフトウェアシミュレータは C で，リダクション方式のソフトウェアシミュレータは Prolog で記述されている．) 4 Queens プログラムの OR 並列に実行についてのシミュレーション結果（台数効果）を図 2.3 に示す．台数を増やしていった時，実行時間が 6〜7 台のところで飽和していることと，4 Queens の OR 並列度が 6.2 であることから，両方式とも Prolog プログラムに内在する並列度を十分に取り出せることが確認された．

なお，リダクション方式ソフトウェアシミュレータで 2 種類の Concurrent Prolog プログラムの AND 並列実行のシミュレーションを行なった結果，リダクション方式 PIM-R が Concurrent Prolog プログラムに内在する並列度を取り出せることが確認された（図 2.4 参照)．

3. 知識ベースマシン

この章では，知識ベースマシンの研究開発について，特にアーキテクチャを中心に記す．

3.1 前期研究課題

前期における知識ベースマシンの研究開発は，中期に開発する知識ベースマシンに必要な基礎的技術を築くこと，並列関係・知識演算が可能なプロトタイプデータベースマシンの基礎研究を行うことを目的として，関係データベースマシンを開発することになった．[Murakami 83]

ICOT における関係データベースマシン（RDBM-Delta-と呼称)開発の具体的目的は二つあり，一つは知識ベース機能をサポートする各種メカニズムとその実現方式を研究する実験環境を実現すること，もう一つは，別途開発中のパーソナル逐次型推論マシン(PSI と呼称)とローカル・エリア・ネットワーク（LAN）[Taguchi 84]

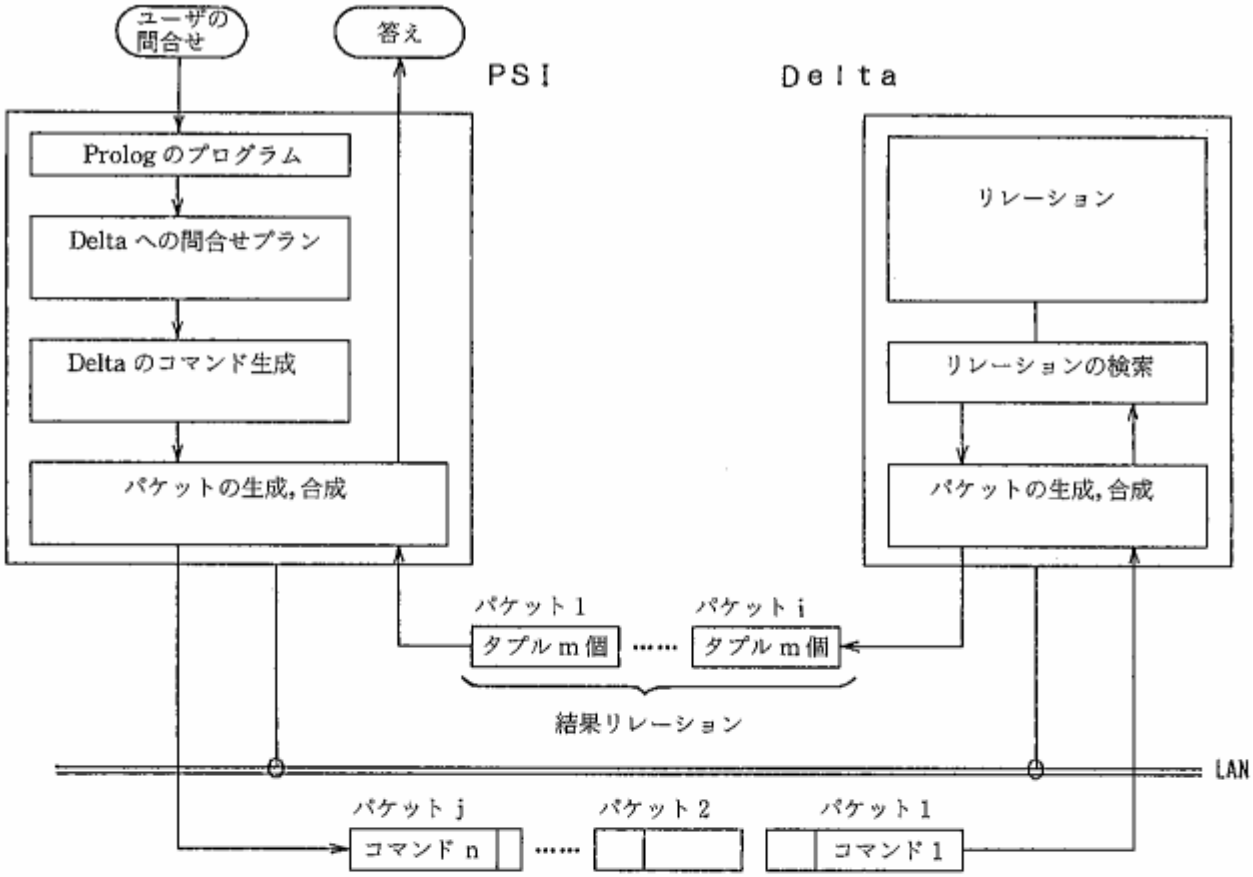


図3.1 PSIユーザの外部データベース問合せ処理フロー

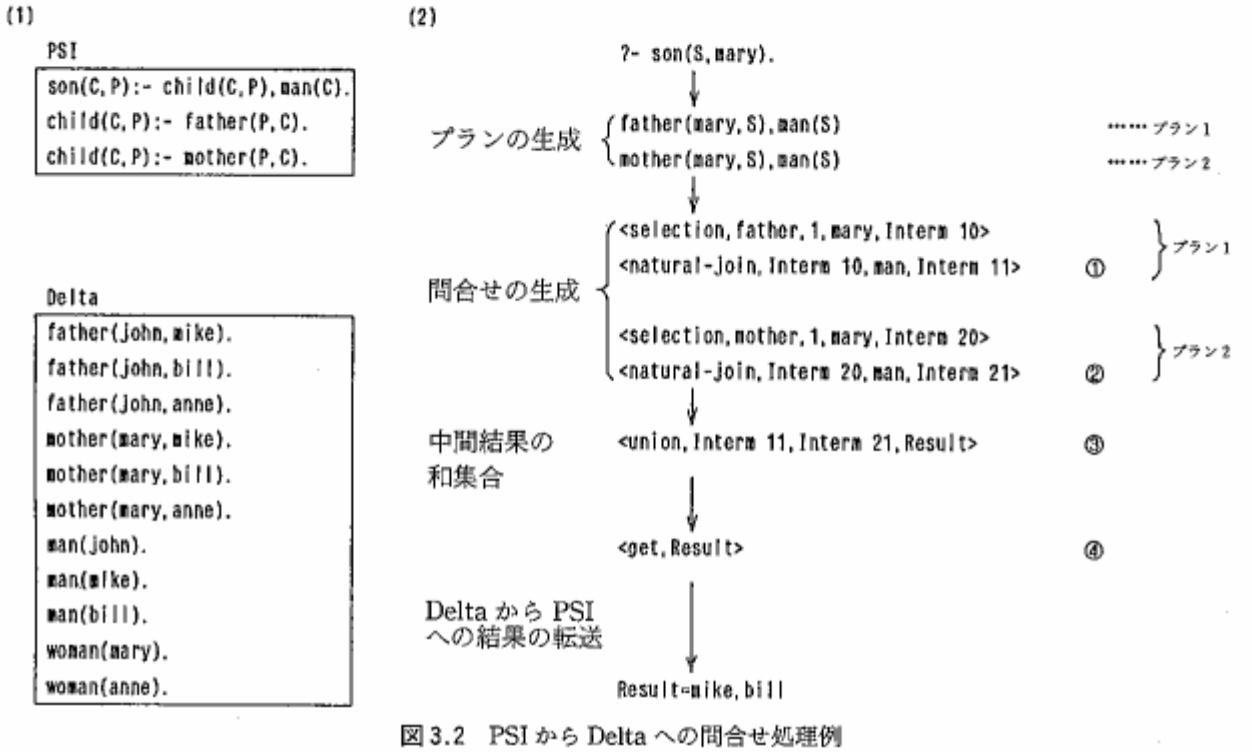


図3.2 PSIからDeltaへの問合せ処理例

を介して接続し，核言語第0版や自然言語などによるソフトウェア開発用ツールを提供することである[Uchida 82]．

3.2 Deltaアーキテクチャ

3.2.1 ホストからの問合せ処理の流れ

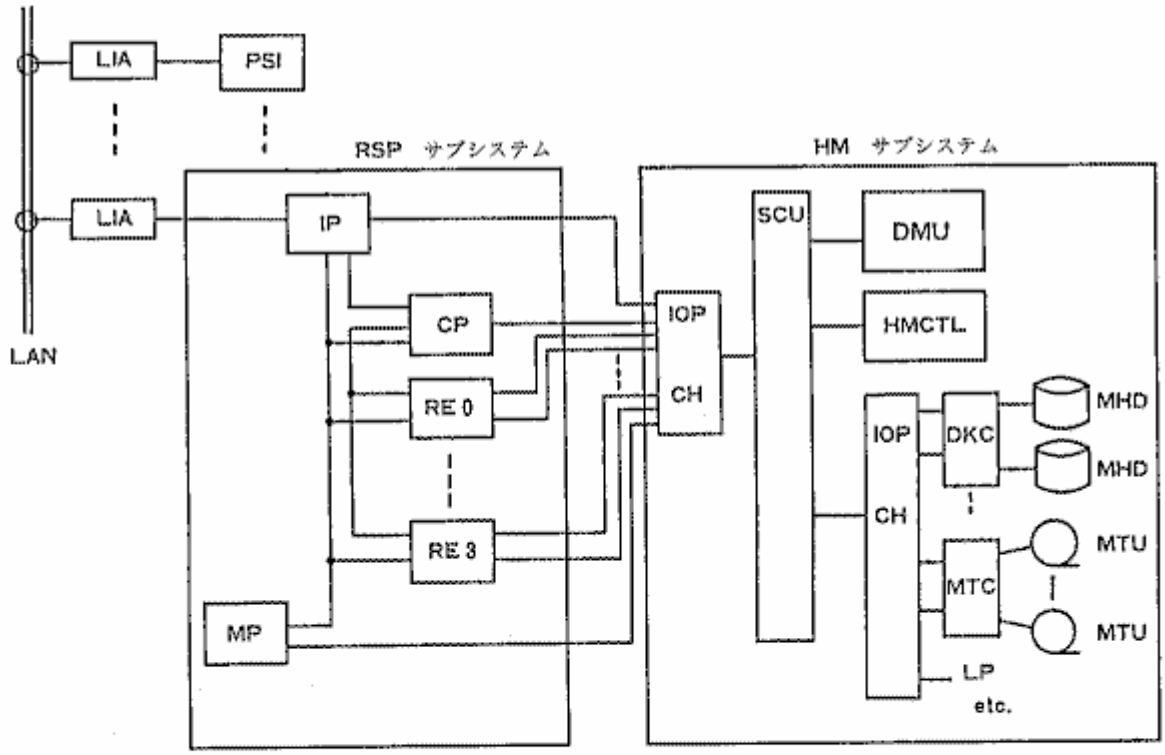
ホストからDeltaへの問合せをしたときの処理の流れを図3.1に記す．ホスト上のユーザはPrologでプログラムを記述し，外部データベースへ問合せを行うものとする．PSIは，ユーザの問合せ内容とPrologプログラムとからDeltaへの問合せプランを生成し，次いでDeltaコマンドに変換し，パケット形成にしたのちLANインタフェースを介してDeltaへ問合せを行う．Deltaはパケット情報からDeltaコマンドを抽出・解析の結果，検索コマンドと判定すると，データベースから所要のデータを検索し，再びLANインタフェースを介してホストへ情報を送出する．

PSIとDeltaの処理の具体例を図3.2に示す．PSIとDeltaには図3.2(1)のようなPrologの節とリレーションが格納されているとする．いま，PSIより?-息子(S, mary)という問を出したときのPSIとDeltaの処理フローを図3.2(2)に示す．PSIはまずプラン1と2を生成し，次いでDeltaコマンド①～④に変換し，Deltaへ送出する．Deltaはデータベースを検索し結果をPSIへ転送する．

この例からもわかるように，Prologで書かれたプログラムの外部データベースへのアクセスの特徴は，次のように考えることができる．

(1) リレーション内の属性（アトリビュート）への平等なアクセスがなされる．

(2) リレーションへの問合せはselection, join,



PSI: personal sequential inference machine, LIA: LAN interface adapter, RSP: RDBM supervisory and processing subsystem, IP: interface processor, CP: control processor, RE: relational database engine, MP: maintenance processor, HM:hierarchical memory subsystem, HMCTL: HM controller, DMU: database memory unit, IOP: I/O processor, SCU: storage control unit, MHD: moving head disk, DKC: disk controller, MTC: MT controller

図3.3 Delta構成

union 等の関係代数演算に変換され，しかも演算負荷の重い join や union の数が多い．

（3） リレーションを構成する属性数は，あまり多くないケースが多い．

3.2.2 アーキテクチャ上の特徴

関係データベースマシンについては，各種の研究がなされているが［Bancilhon 82］，［DeWitt 79］，［Schweppe 82］，［Kitsuregawa 83］，［Tanaka 82］，Delta のアーキテクチャ設定に際しては，3.2.1 節で述べたアクセス特性をも考慮に入れ，以下のアーキテクチャ上の特徴を持たせた［Kakuta 83］，［Shibayama 82］，［Shibayama 83］，［Shibayama 84 a］，［Shibayama 84 c］．

（1） 機能分散型のマルチプロセッサ構成

各プロセッサの制御プログラム作成上の容易さ，実験機としての機能拡張性，データベース処理機能および性能向上のため，5 プロセッサに機能を分散させている．

（2） 関係代数型のコマンドインタフェースの実現

ホストとの論理的コマンドインタフェースとして，関係代数型コマンド（Delta コマンドと呼称）を選択した．また，ホストとのデータ転送形式はタプル型である．上記 Delta コマンドとタプル型データは，LAN プロトコルに従ってホストと Delta 間をパケット形式で転送される．

（3） アトリビュート・ベースの内部スキーマ方式

アトリビュート型の内部スキーマと 2 段階クラスタリング方式を採用した．

（4） 関係代数演算用専用エンジンの具備

関係データベースエンジン（RE）は，パイプライン・2 ウェイマージ・ソート方式による 12 段ソート・セルと関係代数演算処理用マージャとから構成され，最大 4 台の RE が Control Processor（CP）の制御下で並列動作する．

（5） 2 CH ストリーム・データ転送インタフェースの実現

関係データベースエンジン（RE）と階層構造メモリ（HM）との間を，1 RE あたり入力 1 CH，出力 1 CH で独立にストリームデータ転送を行う．1 CH のデータ転送速度は 3 M バイト/秒である．

（6） 階層構造メモリの採用

可動ヘッドディスク（MHD）とデータベースメモリユニット（DMU）の 2 階層構造メモリから成る．DMU は電源障害時の不揮発性を保証させるため Delta システム全体に無停電電源装置を使用している．

（7） 統計情報収集機能の充実

知識ベースマシン開発へ向けて，基礎技術確立のため Delta を使用した実験を行うが，このとき得られる性能情報や各種統計情報の収集機能を Delta に持たせた．

3.2.3 全体構成と諸元

Delta は，全体の制御・監視と演算処理を担当する RSP（RDBM Supervisory and Processing）サブシステムと関係データの格納，検索，変更を担当する HM（Hierarchical Memory）サブシステムとから構成される．図 3.3 に RDBM Delta の構成を示す．

HM と RSP とは，合計 11 本のチャネルインタフェースで接続される．

Delta システムの諸元を図 3.4 に示す．

		1984.5現在の構成	最終構成
(1)RSP サブシステム			
・コントロール・プロセッサ	(CP)	1台	1台
・関係データベース・エンジン	(RE)	1台	4台
・インタフェース・プロセッサ	(IP)	1台	1台
・メインテナンス・プロセッサ	(MP)	1台	1台
(2)HM サブシステム			
・HM 制御装置	(HMCTL)	1台	1台
・データベース・メモリユニット	(DMU)	16 MB	128 MB
・可動ヘッドディスク	(MHD)	5 GB	20 GB
・入出力装置			
磁気テープ装置	(MTU)など	2台	4台

図 3.4 Delta システム諸元

3.2.4 RSP サブシステム構成

RSP サブシステムは，CP，IP，MP，RE から成り，それぞれハードウェアとソフトウェアから構成されている．

（1） RSP ハードウェア構成

各ユニットとも基本部分は 1 プロセッサと 512 KB または 1 MB のメモリとから構成されており，RE ではソータおよびマージャ専用ハードウェアを，また CP ではメモリ容量拡張のため 15 MB の IC バルクメモリを備えている．MP には Delta システム監視用ディスプレイ操作卓，RSP ログ情報収集用 MTU が接続される．RSP の各ユニット間は，3 本の IEEE 488 バスによって相互接続される．

また，HM とのインタフェース用ハードウェアとし

て，CP，IP，MP の各ユニットは各々 1 個の HM アダプタ（HMA）を，RE では入力，出力用に各 1 HMA を備えている．

（2） RSP ソフトウェア構成

RSP の各ユニットのソフトウェアが分担する機能を以下に記す．

（a） CP

（i） トランザクション管理
トランザクション実行管理と Delta コマンド解析，サブコマンド生成・実行を行う．

（ii） 関係データ管理情報の管理

（iii） IP/RE/MP 通信制御

（iv） リカバリ管理
トランザクションを単位としたデータリカバリを行う．[Kakuta 84]

（b） IP

（i） LAN インタフェース制御

（ii） Delta コマンド，データ抽出制御

（iii） コマンド木並列受信制御
コマンド木は，複数個の Delta コマンドから成り，1 つの意味のある処理を行う単位である．本制御ソフトウェアは，受信パケット列からトランザクション毎のコマンド木を識別制御する．

（iv） データ形式変換
ホストからデータ入力時にはタプル識別子（TID）を付加し，逆に出力時は TID を削除等の処理を行う．

（v） HM 間データ転送
CP が IP と HM とのデータ転送用バッファを確保指示，完了後は，IP が HM 間のデータ転送を行う．

（c） RE

（i） RE 制御
CP からのサブコマンドを解析し，エンジン各部のシーケンス制御と HM との入出力動作を制御．

（ii） 関係演算サポート
マージャハードウェアでは処理できない算術演算等の演算を行う．

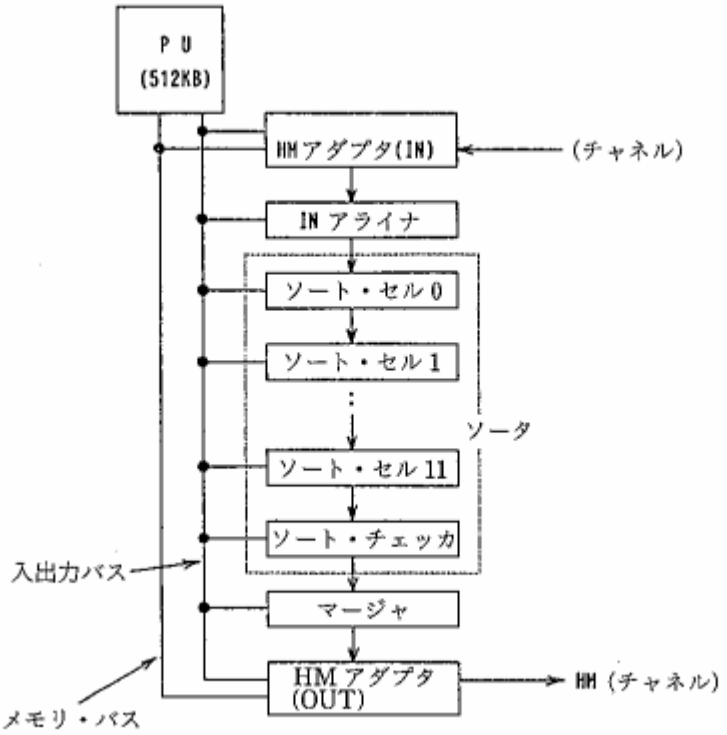


図 3.5 RE の構成

（d） MP

（i） Delta システム状態監視，構成管理

（ii） Delta システム開始，終了制御

（iii） データベースロード・ダンプ制御

（iv） 統計情報収集制御

（v） オペレータコマンド，メッセージ管理

3.2.5 HM サブシステム構成

HM サブシステムは，RSP サブシステムの指示のもとに Delta システムで取扱うデータの格納領域への効率良い書込み，および読出し管理を行うための HM ハードウェアと HM ソフトウェアとから成る．

（1） HM ハードウェア構成

HM ハードウェアは，最終構成で 128 M バイトの記憶容量を持つ不揮発化した高速 DMU と最大容量が 20 G バイトの磁気ディスク装置とこれを制御するディスク制御装置と HM 制御プログラムの実行を担当する HM コントローラなどから成る．

（2） HM ソフトウェア構成

HM ソフトウェアが実行する機能を以下に記す．

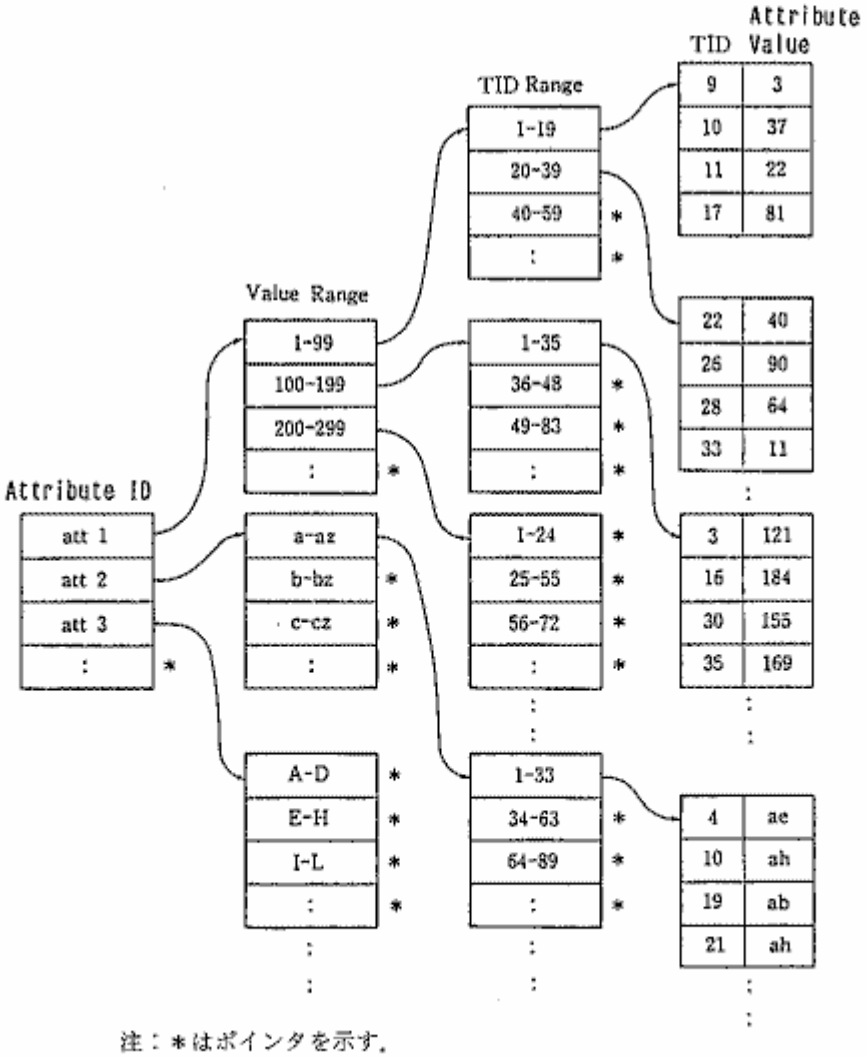
（a） HM サブコマンド処理

RSP から指示された HM サブコマンドにより以下の処理を行う．

（i） アトリビュート定義，操作

アトリビュート定義の生成，消滅およびタプル再構成，アトリビュート分割処理を行う．

（ii） 更新処理



注：＊はポインタを示す．

図 3.6　Delta の内部スキーマ

アトリビュートに対する挿入，削除，更新処理を行う．

(iii) クラスタリング操作

(iv) データ転送管理

HM-RE 間のストリーム・データおよび HM-IP，CP 間のパケット・データの転送処理を行う．

(v) RSP バッファ管理

HM 内 RSP バッファの確保，リリース処理を行う．

(b) メモリ・リソース，MHD スペース管理

アトリビュート情報，ディレクトリ情報の DISK と DMU 間転送制御と，これらの情報を格納するメモリ・リソースの管理を行う．

(c) データリカバリ処理

CP からのリカバリ系サブコマンドにより，アトリビュート情報のリカバリ処理を行う．

3.3 関係データベースエンジンの処理方式

RE はソータとソート結果を比較演算するマージャとから構成されるが，その主要構成要素であるソータには，2 way マージ・ソートアルゴリズム [Knuth 73] を採用した．

(1) 2 way マージ・ソートアルゴリズム

本アルゴリズムは，入力レコードを値の昇順または降順に並べかえを行うもので，入力レコード列を長さが 1 のソートされた列が並んだものとみなし，これを 2 つずつマージして，ソート列の長さを各ソート段で倍々にしていく．この操作をパイプライン処理することにより，レコード数を n としたとき，$\log_2 n$ 段のソータを使用することにより，$2n+(\log_2 n)-1$ 回のサイクルタイムでソート出力が可能となる [Todd 78]．

(2) RE の構成 [Oka 84]

RE は，IN モジュール，12 段ソート・セルからなるソータとマージャと 2 個の HM アダプタとこれらを制御するプロセッサと RE 制御プログラムとから成る．RE

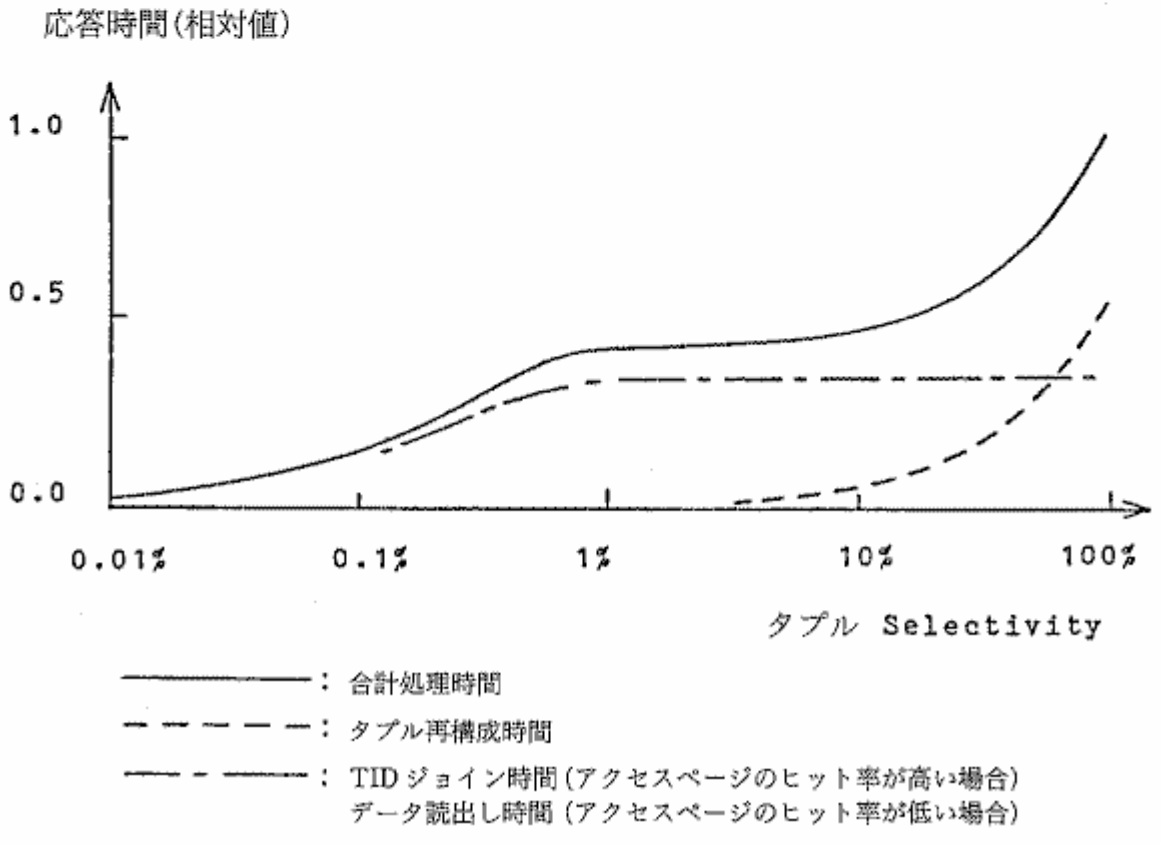


図 3.7 Delta の性能予測特性

の構成を図3.5に示す．

REの関係代数を基本とする演算は，アトリビュート値とTIDを組にしたアイテム群をストリーム状に転送しながらソートし，ソート出力の先頭から比較演算を行い，その結果に基づいたアイテムの組合せを出力する[Iwata 84]，[Sakai 84]．

INモジュールでは，各アイテムのソート対象部分をその先頭におきかえる操作を行う．ソート・セルは2個のFIFOバッファと比較器とマージ用マルチプレクサとから成り，FIFOバッファ容量は，第1段が各々16バイト，第12段が各々32Kバイトである．

マージャは，2個の64KバイトFIFOバッファと比較演算部と出力制御部とからなり，2ストリーム内のアイテムの比較対象部分の組合せ出力を比較演算結果によって決定し，HMAを経由してHMに送出する．

3.4 内部スキーマと記憶空間管理方式

3.4.1 内部スキーマ

データベースマシンの性能に大きく影響する内部スキーマには，タプル型格納方式とアトリビュート型格納方式とがあるが，Deltaではリレーションを構成するタプルをアトリビュートに分割し，アトリビュート毎に格納する後者の格納方式を以下に示す理由から採用した[Miyazaki 83]．

（1） Prologをはじめとする論理型プログラミング環境下でのDeltaへのアクセスパターンは，どのアトリビュートも平等にアクセスされる可能性が大きいこと．

（2） 問合せに対し，必要なアトリビュートのみアクセスすればよいこと．
これに対しタプル型格納方式の場合は不必要とするアトリビュートもアクセスの対象となってしまう欠点がある．

（3） マージ・ソートアルゴリズムで実現したREの処理効率，性能上適したデータ格納方式であること．

しかし，アトリビュート型格納方式を採用した結果，次に示すデメリットもある．

（a） タプル再構成とアトリビュート分割処理が必要であること．

（b） タプル識別子(TID)を各アトリビュート毎に付してタプルを識別させるため，記憶空間の有効使用が悪いこと．

Deltaではアトリビュート情報のサーチスペースを削減するため，アトリビュート識別番号（Attribute Identifier），アトリビュート値の範囲とTIDの値の範囲をサーチキーとする2段階クラスタリングを採用した．図3.6にDeltaの内部スキーマを示す．TIDと値のペアで構成されるアトリビュート情報は，アトリビュートIDにより他のアトリビュートと区別され，まずアトリビュート値でソートされたのち，特定の値の範囲ごとに分割し一時的にワーク用ページ空間に格納する．次に，この各ページ空間の情報をTIDでソートし図3.6右端に示すTID—アトリビュート値ペアから成るページ空間群に格納し，次いでアトリビュート値の範囲とTID Range Tableとのポインタを示すAttribute Range Tableと，TIDの値の範囲とTID—Attribute値とからなるページ空間とのポインタを示すTID Range Tableとを作成する．

3.4.2 空間管理方式

HM内のMHDには，内容に応じデータが3つのデータセットに分割して格納される．

（1） RSPデータセット
ディレクトリと呼ばれるデータベース管理情報であり，その内容はCPが管理する．

（2） アトリビュート・データセット
アトリビュート情報が格納される空間である．

（3） RSPページ・データセット
HM-RSP間データ転送時RSPバッファ空間不足のためあふれた情報を，一時的に退避するために設けたHM上の空間である．

3.5 性能予測

Deltaの性能を予測のため，以下の条件でSelectionの問合せをDeltaに行ったときのタプルSelectivityと実行時間特性を図3.7に示す[Shibayama 84a]，[Shibayama 84b]．

条 件

（1） リレーションは，タプルあたり10アトリビュートで10000個のタプルから構成される．

（2） 各アトリビュートは10バイトである．

4. おわりに

前期におけるハードウェアに関する研究・開発は，並列推論マシンおよび知識ベースマシンについて独立に進

めている．

並列推論マシンに関しては，それが動作する環境条件を解析し，データフローリダクション，節単位処理および完全コピーの4方式についてアーキクチャの設定を行い，ソフトウェアシミュレータの開発あるいは実験機の試作を行っている．

今後は，

（1） 各種並列推論マシン方式について，より詳細なソフトウェアシミュレーション

（2） 各種並列推論マシン方式の評価のための並列プログラムの検討

（3） 各種並列推論マシン方式の実験機（モジュール数8台～16台）による各種実験評価

を行う予定である．

知識ベースマシンに関しては，関係データベースマシンDeltaを開発中であり，59年5月末両サブシステムのハードウェアの接続テストを完了した．現在，両サブシステムのソフトウェア統合試験を実施中である．

また，推論機構と関係データベースマシンとの関係などについても検討を進めている．[Yokota 83]，[Yokota 84a]，[Yokota 84b]

今後は，

（1） Deltaハードウェアの全体システムの実現と，ソフトウェアの機能拡充

（2） LAN経由でPSIを結合した動作環境での，実データの収集，および有効性の評価

（3） 最大4台のREを用いた，ストリーム多重化処理アルゴリズムの実験

（4） PSI-Delta密結合による実験

などを行う予定である．

これらの前期における研究・開発の進歩をもとに，中期におけるハードウェアの研究・開発を，次の様に展開する．

上に述べた前期研究を強力に推進し，各種評価データの収集とこれに基づく各方式の有効性を定量的に評価する．

次に，この検討をもとに，並列推論マシンおよび知識ベースマシンのアーキテクチャを設定し，このアーキテクチャを実現するハードウェア構成技術を開発する．

更に，上記の検討と同時に，いくつかの応用プログラムが動作する，上記マシンを中核とした推論サブシステムおよび認識ベースサブシステムを構成する．

推論サブシステムと知識ベースサブシステムの結合方式の検討は，各サブシステムの立場から相手サブシステムをどの様に統合するかを独立に検討し，この双方の検討結果をもとに最適な結合方式を明確にする．

謝　辞

本研究・開発は，ICOT研究所の第1研究室と担当メーカとの緊密な協力のともで行なわれた．関係各位の御努力に感謝したい．また，WG1のメンバの方々には重要な問題点について貴重な御意見をいただいた．心から謝意を表する．

〈参　考　文　献〉

[ICOT 84] 電子計算機基礎技術開発成果報告書　推論サブシステム編，(財)新世代コンピュータ技術開発機構，1984

[Ito 83] Ito, N. Onai, R. et al.: Prolog Machine Based on the Data Flow Mechanism, ICOT Technical Memorandom TM-0007, 1983

[Ito 84] Ito, N. and Masuda, K.: Parallel Inference Machine Based on the Data Flow Model, International Workshop on High-Level Computer Architecture 84, 1984

[Onai 84-a] 尾内，他：逐次型Prologプログラムの解析，Logic Programming Conference 84，1984

[Onai 84-b] Onai, R. and Asou, M.: Parallel Inference Machine Based on Reduction Mechanims-its architecture and software simulation, ICOT Technical Report TR-077, 1984

[Shapiro 83] Shapiro, E. Y. (Weizmann Institute): A Subset of Concurrent Prolog and Interpreter, ICOT Technical Report TR-003, 1983

[Bancilhon 82] Bancilhon, F. et al. VERSO: A Relational Back-End Data Base Machine, Proc. of Int'l Workshop on Database Machines, Aug. 1982.

[Dewitt 79] Dewitt, D. DIRECT-A Multiprocessor Organization for Supporting Relational Database Management Systems, IEEE Trans. on Computers, C-28, No. 6, June, 1979.

[Iwata 84] Iwata, K. et al. Design and Implementation of a Two-Way Merge-Sorter and its Application to Relational Database Processing, ICOT Technical Report TR-066, May 1984.

[Kakuta 83] Kakuta, T. et al. RDBM Delta (I), (II) and (III), Proc. of 29th National Conf. of IPSJ, Mar. 1983 (in Japanese), and also ICOT Technical Memorandum TM-0008. (in English)

[Kakuta 84] 角田，他：RDBM Deltaのリカバリ方式，情処学会第29回全国大会，1984. 9

[Kitsuregawa 83] Kitsuregawa, M. et al. Application of Data

Base Machine and Its Architecture, New Generation Computing, 1, No. 1, 1983.

[Knuth 73] Knuth, D. E. et al. Sorting and Searching, The Art of Computer Programming,Vol. 3, Addison-Wesley Publishing Co., 1973.

[Miyazaki 83] 宮崎，他：データベースマシンにおけるデータ格納法に関する一考察，情処学会第27回全国大会，1983.10

[Murakami 83] Murakami, K. et al. A Relational Database Machine: First step to Knowledge Base Machine, Proc. of 10th Symposium on Computer Architecture, Stockholm, Sweden, June 1983. Also available as ICOT TR-012.

[Oka 84]岡，他：関係データベースエンジンの開発，情処学会第29回全国大会，1984.9

[Sakai 84] Sakai, H. et al. Disign and Implementation of the Relational Database Engine, Proc. of Int'l Conf. of Fifth Generation Computer Systems 1984, Nov. 1984, and also ICOT TR-063.

[Schweppe 82] Schweppe, H. et al. RDBM-A Dedicated Multiprocessor System for Data Base Management, Proc. of Int'l Workshop on Database Machines, Aug. 1982.

[Shibayama 82] Shibayama, S. et al. A Relational Database Machine "Delta", ICOT TM-0003, Nov. 1982.

[Shibayama 83] Shibayama, S. et al. On RDBM Delta's Relational Algebra Processing Algorithm, Proc. of 27th National Conf. of IPSJ, Oct. 1983, and also ICOT TM-0023.

[Shibayama 84a] Shibayama, S. et al. A Relational Database Machine with Large Semiconductor Disk and Hardware Relational Algebra Processor, New Generation Computing, 2, No. 2, 1984.

[Shibayama 84b] Shibayama, S. et al. Relational Database Processing on an Attribute-based Schema, Proc. of 29th National Conf. of IPSJ, Sep. 1984.

[Shibayama 84c] Shibayama, S. et al. Query Processing Flow on RDBM Delta's Functionally-Distributed Architecture, Proc. of Int'l Conf. of Fifth Generation Computer Systems 1984, Nov. 1984, and also ICOT TR-064.

[Taguchi 84] Taguchi, A. et al. INI: Internal Network in ICOT and its Future, Proc. of 7th Int'l Conf. on Computer Communications, Oct. 1984.

[Tanaka 82] Tanaka, Y. A Data Stream Database Machine with Large Capacity, Proc. of Int'l Workshop on Database Machines, Aug. 1982.

[Todd 78] Todd, S. Algorithm and Hardware for a merge Sort Using Multiple Processors, IBM J. of Res. Develop. Vol. 22, No. 5, Sep. 1978.

[Uchida 82] Uchida, S. et al. The Personal Sequential Inference Machine Outline of Its Architecture and Hardware System, ICOT Technical Memorandum, TM-0001, Nov. 1982.

[Yokota 83] Yokota, H. et al. An Investigation for Building a Knowledge Base Machine, Proc. of 27th National Conf. of IPSJ, Oct. 1983 (in Japanese), and also ICOT TM-0019. (in English)

[Yokota 84a] Yokota, H. et al. An Enhanced Inference Mechanism for Generating Relational Algebra Queries, Proc. of 3rd ACM SIGACT-SIGMOD Symposium on Priciples of database systems, Apr. 1984, and also ICOT TR-026.

[Yokota 84b] Yokota, H. et al. Unification in a Knowledge Base Machine, Proc. of 29th National Conf. IPSJ, Sep. 1984.