

## (2) アーキテクチャとハードウェアシステム; 並列型推論マシン (PIM) と知識ベースマシン (KBM)

ICOT 第1研究室長 村上 国男

ICOT 第一研究室長の村上でございます。

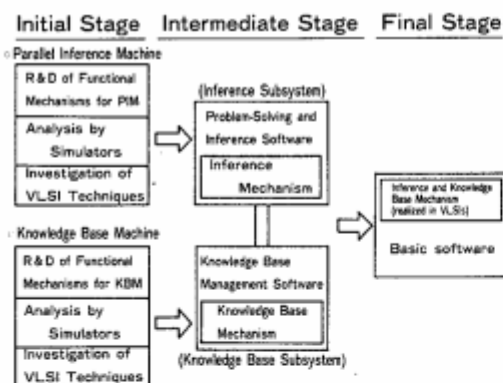
ただ今より、アーキテクチャ及びハードウェアシステムと題しまして、1982年から約2年半にわたってすすめてまいりました並列推論マシン及び知識ベースマシンの、主にアーキテクチャサイドの検討に関しましてご報告を申し上げます。なお、今回ここで発表致します内容は、現在精力的に研究開発を展開しております各マシンについての中間結果でございます。前期3年の研究開発の終了時点、来年の4月までには、より定量的なデータに基づいた評価結果をお示しできるのではないかと考えております。

ち前期3年間の研究開発を遂行中でございますが、この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実施致しまして、中期に実現する各々のサブシステムの基本的な構成技術を確認することでございます。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システム構成要素でございます。最終的なシステムでは、恐らく、この2つのマシンの機能は相互に密に結合されると予想されております。しかし、前期では、FGCSを構成する個々の構成要素の研究開発が主でございます。中期で構成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認するということが目的でございます。このために前期3年間のハードウェア、アーキテクチャの研究開発では、並列推論マシンと、知識ベースマシンの各々の研究項目を独立に設定いたしました。現在、前期研究開発では、各々のマシンに関しまして、主に次の3つの視点から研究を展開しております。

第一は各マシンの構成要素メカニズムを機能的な側面から検討致しまして、マシンアーキテクチャとしての基本的に具備すべき機能を明確に抽出することでございます。

第二は、これらの機能を実現する各種方式に関し、適応性の限界等を各種シミュレーション、あるいは、実験機の試作等を通じて評価データの収集・解析を行い、定量的に把握することでございます。

第三は、重要なハードウェア的な基本構成要素に関しましては、VLSIの試作を通じまして、実現



Stages of Architecture Research & Development

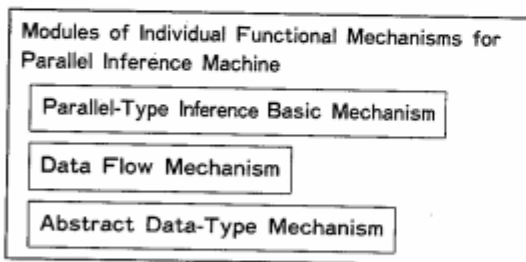
図1

さて、第五世代コンピュータシステムに関する研究開発の最終目標は、図1に示すように、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ことのできる知識情報処理システムの、パイロットモデルを開発することでございます。ご承知のように、このFGCSプロトタイプの実現のために3期の研究ステップが設定されました。現在このう

の可能性とその構成手法を明確にしようということとでございます。

前期に於ける並列推論マシンの研究開発は、図2に示しました3つのメカニズムの評価、検討を主たる目標としております。現時点に於けるこの3つの課題に關します研究開発状況は次の通りで

Initial Stage: Development of Basic Computer Technology



#### Research & Development Items of PIM

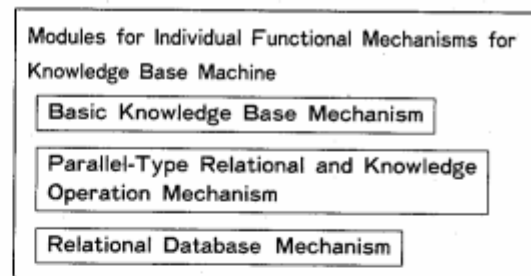
図2

ございます。第一の並列型推論基本メカニズムの研究開発に關しましては、基本メカニズムと致しまして、リダクション、節単位処理、完全コピー方式の3方式を選択致しまして、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及びソフトウェアシミュレーションによる評価を行なっておりまして、現在シミュレーション専用装置の試作を行なっている状況でございます。中期前半に於きましては、これらの試作機によりまして、各種のデータを収集致しまして、次に申し上げますデータフローメカニズムとあわせて比較検討を行ない、本来、並列推論マシンの方式として何が有効であるかということを確認にしたいと思っております。第二のデータフローメカニズムに關しましては、ソフトウェアシミュレーションによる評価を行いつつありまして、現在この評価データ等を参考にデータフローメカニズムに基づいた並列推論マシンの実験機を試作中でございます。中期前半に於きまして、やはりこの試作機を用いて各種データを収集したいと考えております。これらの結果

を総合し、データフローメカニズムの並列推論マシンに対する適用限界を明らかにしたいと考えております。第三の抽象データ型メカニズムに關しましては、むしろ言語の機能からの検討が先行すべきであると判断しまして、現在、核言語第一版の機能の検討の中で概念の整理と機能の抽出とを行なっております。

前期における知識ベースマシンの研究開発は、同様に図3に示した3つのメカニズムに關しまして、各々の実行モデル、アーキテクチャとして具備すべき機能、及びハードウェア実現方式等の検討を行なうことを主目的に致しました。現在、第一の知識ベース基本メカニズムに關しては、知識ベース演算についての知見や評価データ等が、アーキテクチャ確立の研究を展開するには、まだまだ不十分であると判断致しました。この為、前期

Initial Stage: Development of Basic Computer Technology



#### Research & Development Items of KBM

図3

に開発する関係データベースマシン Delta 等を使用致しまして、各種実験による知見の十分な蓄積を計った後に、研究を展開すべきであると考えておりまして、中期の初頭、60年の初頭から研究を大々的に展開しようと計画しております。第二の並列型関係知識演算メカニズムの研究開発に關しましては、並列型関係メカニズムにつきましては、複数台の逐次推論マシンと先程申し上げました Delta とを LAN インターフェースによって比較

的疎に結合し、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環境を確立致しました。さらに、逐次型推論マシンPSIと関係データベースマシンDeltaとを、LANを介さないで直接密に結合するような動作環境を現在実現中でありまして、この密結合による実験データの収集・評価、及び知識演算メカニズムそれ自身の研究開発に関しましては、中期のやはり前半から大々的に展開したいと考えております。第三の関係データベースメカニズムに関しましては、第一に知識ベースマシンに向けての基礎的な技術確立するための実験ツールと、それから第二にはICOTにおけるソフトウェア開発支援システムを構築するための構成要素、これを提供することを目的と致しまして、大容量のデータベースマシンDeltaを現在開発中でありまして。

以上が各マシンの研究開発の目的と進捗の現状についての概要でありまして、以下に並列推論マシンと知識ベースマシンに関しましてやや詳しく現在の状況をご報告致します。なお、技術的な詳細の内容につきましては、時間の都合もございまして、この本会議の後半、テクニカルセッションで研究発表を行っておりますので、あわせて御参考にしていただければと存じます。

PIMに関しまして前期研究の目標は、並列推論マシンの動作する環境を十分に解析致しまして、この結果をマシンの設計条件に反映させること、及びマシン実現のための構成方式を明解にするこ

とでありまして、図4に示す2つのステップで検討をすすめることに致しました。第一は、核言語の特性解析でございますが、第五世代コンピュータに於ける核言語の位置は、ソフトウェアとハードウェアとのインターフェースを規定するものでございまして、マシンアーキテクチャの設定に決定的な影響を与える立場にございまして。よって核言語で記述されたプログラムが、いったいどのような特性を持ち、どのような振る舞いをするかということ十分に解析致しまして、定量的に把握したのちに、PIMのアーキテクチャ検討をすすめるべきであると考えました。そこで、第一に核言語のベースランゲージでございます Prolog 及び Concurrent Prolog で記述されたプログラムの静的及び動的特性の解析を行ない、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べきであろうと考えております。第二のステップは、今まで提案されております各種並列方式の検討を行なうこととございまして。各種並列方式の特性とそれらの方式に基づく並列推論マシンアーキテクチャの検討を行なったのちに、ソフトウェアシミュレーションによって各方式の有効性の検討、定量的評価、あるいは実験機試作等を通じまして、マシンの構成技術に関する中期に向けた基本的な知見を確立するというステップとございまして。

核言語の特性解析に関しましては、核言語のベースランゲージのひとつであります Prolog で記述されたプログラムの静的な、あるいは動的な解析を行ないました。図5に示しましたのは、静的解析結果のひとつの例でありまして、例えば平均の参照数が3であるとか、あるいはヘッドリテラルの平均の引き数の個数は3であるとか、今まで逐次型の言語でございます Prolog に関しましては、ICOTで開発しましたプログラムを中心と致しまして、いくつかのデータを得ております。

図6は動的解析結果の一例でありまして、まず今申し上げました Prolog を除去するとか、否定

### Parallel Inference Machine (PIM)

#### 1 Analysis of kernel language

Analysis results → Architecture of PIM

#### 2 Comparing various parallel inference mechanism

- Evaluation using software simulators
- Development of experimental machines with 8 to 16 modules

図4

## Results of Static Analysis

Static analysis of inference clauses  
33 Prolog programs (consisting mainly of inference clauses) developed at ICOT were analyzed.

The data obtained is:

Average number of cuts per clause	0.65
Average number of references	3
Average number of arguments in a head predicate	3
Ratio of structure data to head predicate arguments	20%
Ratio of evaluable predicates to literals	50%
(Ratio of evaluable predicates not requiring unification)	25%

図 5

演算を導入するというような条件を追加致しまして、並列実行可能な Prolog プログラムに変換し実行させまして、その過程で、いくつかのデータを収集致しました。まず、この例でおわかりになりますように、動的な OR のリレーション数は、5 ないし 7 程度が現在我々が手元にあるプログラムの特性でございます。また、構造体データの処理

## Results of Dynamic Analysis

	Morphological analysis	Predicate logic formula simplification
Number of dynamic OR relations	7.1	4.6
Ratio of structure data	32%	50%
Ratio of evaluable predicates	40%	80%
Degree of OR parallelism	8.3	3.2

図 6

が、全体のプログラムの実行に関して、30 ないし 50% にも及ぶということが明らかになっております。現在更に大量にデータを収集中でございますが、今までの静的および動的な解析結果を総合致しますと、だいたい次の 3 つのことが明らかになったのではないかと考えております。まず第一は、

構造体データの処理が非常に重要であると考えられます。このためにストラクチャメモリの処理メカニズムに関しましては、重点的な研究の展開が必要であろうと考えております。第二は引き数の個数が現在我々手もちの Prolog プログラムでは比較的少なく、引き数間の並列度はたかだか 5 個程度であると言ってよろしいのではないだろうかと考えております。第 3 は、プロセッシングエレメントにおける Built-in-predicate の実行、いわゆるディターミニスティックな処理に関しまして、この高速化というものが全体のマシンの高速化に影響を与えるということが言えるのではないかと思います。もちろん、今まで解析いたしましたのは、本格的な並列 Prolog のプログラムではございませんで、逐次型 Prolog を人為的に並列に実行できるようなかたちに整えまして、それで実行したものでございますので、アルゴリズムそれ自身を、最初から並列にくみあげるところの要因は無視しております。ですから今後、本格的な並列プログラムを実行いたしまして、これらの動的及び静的な解析を行なったのちに、プログラムの本来の動作特性を把握しなければいけないと考えております。

さて、並列推論マシンのマシンアーキテクチャに関してでございますが、図 7 に示す通り、ただ今 4 つの方式につきまして研究をすすめております。最初のリダクション方式、それから 2 番目のデータフロー方式は、後程マシンの構成等をお示ししたいと思います。

## Machine Architecture

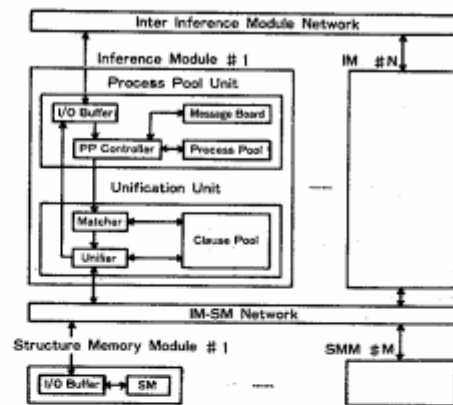
- (1) Reduction Mechanism
- (2) Dataflow Mechanism
- (3) Complete-copying Mechanism
- (4) Clause-Unit-Processing Mechanism

図 7

3番目の完全コピー方式、及び4番目の節単位処理方式に関しましては、時間の都合で簡単に各々の位置づけを申し上げるにとどめておきます。完全コピー方式はリダクション方式のひとつと見なすことができる方式でございますが、並列を極限まで追求するという立場のモデルでございます。リダクション可能なリテラルを含むプロセス全体をすべてコピーして、他とは全く切りはなしてしまっ、そのプロセスだけで独立に処理を実行できるというモデルでございます。このため各プロセスの独立性が高くなりますが、ややコピーの為のオーバーヘッドが大きくなり、かつネットワークを通るパケットのデータ量が多くなるという欠点があるかと思ひます。節単位処理方式

節単位処理方式はこれはむしろシークエンシャルなマシンの自然な拡張というように考えておきまして、むしろマルチプロセッサのモデルと申しあげた方がわかりいいのではないかとお思います。これは各処理ユニットでは深さ優先で処理を行ないまして、暇な処理ユニットから仕事の分割の要求があった時だけ、自分のところで株分け可能な最大の仕事を要求したプロセッシングにわけ与える、という処理方式でございます。こういうところから資源要求の爆発や、並列化のオーバーヘッドが防げるというメリットがございます。以上4つの方式につきまして、現在研究をすすめておきます。

図8は、リダクション方式に基づくPIM-Rと呼んでおりますマシンの構成でございます。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、ゴールとクローズからのレゾルバントの生成の過程でございます。リダクションの過程と非常に親和性があると考えております。図に示したとおり、インフェレンスモジュールとストラクチャメモリーモジュール、及びこれらを結合するネットワークからなっております。インフェレンスモジュールは、プロセスプールユニットとユニフィケーションユニ



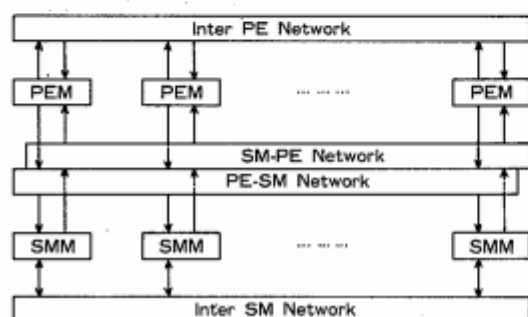
### Conceptual Configuration of PIM-R

図8

ットからなっておりまして、プロセスプールユニットは、プロセスとプロセス間の関係の情報が格納されております。リダクション可能なプロセスはコントローラによって撰択された後に、ユニフィケーションユニットに送られます。また、インフェレンスモジュールの中には、Concurrent Prolog用にメッセージボードを設けておきまして、プロセス間のチャンネル、及びチャンネルを通じて送られる値と、この値の到着を待つサスペンドプロセスを格納しております。ユニフィケーションユニットは御存知の通り、ユニフィケーションを実行するユニットでございます。ストラクチャメモリーモジュールは構造体データを格納いたしましてユニフィケーションに必要な構造体データをユニフィケーションユニットに転送する役目を背負っております。このアーキテクチャの特徴は、プロセスプール内のプロセスはひとつ以上のポディリテラルから構成されておりますけれども、本方式ではリダクション可能なリテラルのみをコピーする、いわゆる部分コピー方式を採用している事でございます。この結果、プロセス全体をコピーするコピーのオーバーヘッド、あるいはネットワークを通過するデータ量の削減が計られると考えております。また、本方式では核言語第一版のベースランゲージでございます Prolog を OR

並列に、Concurrent Prologを主としてand並列に実行する機能を持ち、KLIを統一的に実行するモデルになっております。

図9は、データフロー方式に基づいたPIM-Dと呼んでおりますマシンの構成を示しております。データフローマシン方式は処理に必要なデータがすべて揃った時点で、それが引き金となって実行が開始されるという概念でございます。プログラムに内在する並列性をプログラマーが陽に指定しなくても、十分にひきだす事が可能な方式でございます。この概念に基づきまして、格言語



Conceptual Configuration of PIM-D

図9

プログラムをOR並列に、あるいはAND並列に、そしてユニフィケーション処理も並列に実行するというマシンの実験機を開発中でございます。ここに示したように、本マシンは推論の基本処理を行なうプロセッシングエレメントモジュールと、構造データの格納、あるいは管理を行なうためのストラクチャメモリーモジュール、及びこれらを結ぶネットワークとから構成されております。プロセッシングエレメントモジュールは命令の実行制御、手続きの呼びだし制御、及び基本的なパターンマッチング機能および組み込み述語関数の実行を受けもっております。これはインストラクションコントロールユニットとアトミックプロセッシングユニットのふたつの部分からなっておりますが、インストラクションコントロールユニット

は結果バケットの到着した時に起動されます。この結果バケットをもとに致しまして、宛て先で指定された命令の全オペランドが揃っているかどうかを判定し、命令が実行可能になったのであればアトミックプロセッシングユニットに処理を渡すという構成になっております。ストラクチャメモリーモジュールは、リダクション方式のところでご説明したと同様の処理を行なうユニットでございます。このアーキテクチャの特徴は、第一に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、and並列、それからユニフィケーション処理にあらわれる並列処理、この3つの並列処理をマシン上で統一的に実現しているということでございます。

第二の特徴は、ユニフィケーション処理を行な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でございます。すなわち構造データは、ストラクチャメモリーモジュール群に分散して格納されておまして、プロセスを実行するプロセッシングエレメントモジュール群間ではそのアドレス、いわゆるポインタだけを受け渡しておりまして、プロセッシングエレメントモジュールは構造データアクセスが必要な時にだけSMMにアクセスするという方式をとっております。これによりまして、複雑な構造データのコピーのオーバーヘッド、あるいはこれを転送するためのオーバーヘッドが削減されまして、並列処理で問題になる各種のオーバーヘッドが部分的に解消されているのではないかと考えております。

次に、図10に図すようなソフトウェアシミュレーションによりまして、2つの方式を評価いたしましたので、そのデータを御紹介したいと思います。データフロー方式、及びリダクション方式に関しまして、いくつかのプロセッシングエレメントが並列実行するモデルをシミュレートするソフトウェアシミュレータを開発いたしました。データフロー方式のソフトウェアシミュレータはCで書かれておまして、リダクションの方式の

### Evaluation through Software Simulation

- Software simulator for PIM-R — written in Prolog on DEC 2060 and vax11/780
  - ・ OR parallel execution of Prolog programs
  - ・ AND parallel execution of Concurrent Prolog programs
- Software simulator for PIM-D — written in C on vax11/780
  - ・ OR parallel / stream AND parallel execution of Prolog programs

図 10

おります。これを用いました実験の結果を、以下にご報告致します。

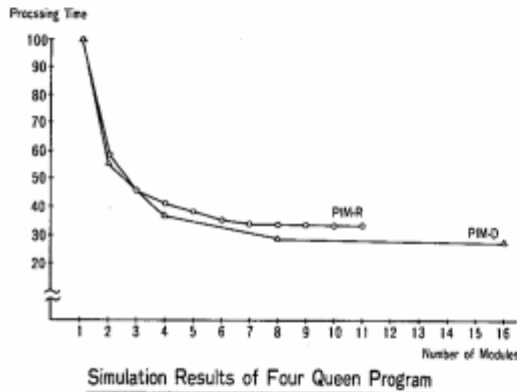


図 11

まず、4-Queen プログラムを OR 並列に実行したシミュレーション結果を図11に示しました。この4 Queen プログラムは Prolog で記述されておりまして、台数を増やしていった時に、このグラフにありますとおり実行時間が6ないし7台のところまで飽和しております。4 Queen のOR並列度がダイナミックなアナライザで解析した結果、だいたい6ないし6.5程度でございますから、この結果、両式とも Prolog プログラムに内在する、少なくとも4 Queen プログラムで示されましたような並列度は十分にひきだしているということがいえると思います。

Concurrent PrologプログラムのAND並列実

行に関しまして、リダクション方式をソフトウェアシミュレータでシミュレーションを行ないました。その結果が図12でございます。ダイナミックアナライザの解析では Integer generator の並列度が2でございます、10個の Quicksort でだいたい6程度でございます。そう致しますと、図11の議論と同じくこの図からリダクション方式が Concurrent Prologのプログラムに内在されております並列度を十分にひきだしている、少なくとも、今後ひきだす可能性は十分にある方式であるということがいえるのではないかと考えております。以上が並列推論マシンの中間結果の報告でございます。

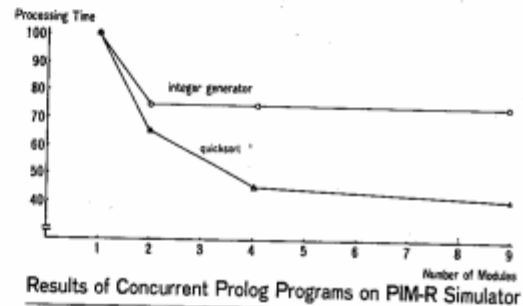


図 12

次に知識ベースマシンの研究開発の御報告にうつらせていただきます。前期に於ける知識ベースマシンの研究開発は、図13に示しました通り、知

### Knowledge Base Machine (KBM)

1. Development of relational database machine
  - ・ an experimental environment in which various knowledge base functions can be investigated
  - ・ a software development tool connecting Delta with PSIs via LAN.
2. Data collection toward KBM using Delta
  - ・ experiments of Delta tightly-coupled with PSI.
  - ・ parallel processing experiments of queries using up to four REs.

図 13

識ベースマシンに必要な基礎的な技術を明らかにすること、及び並列関係、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なうこととございまして、関係データベースマシンを現在開発中とございます。ICOT に於ける関係データベースマシン、Delta と我々は呼んでおりますが、この開発の具体的目標は2つとございます。

1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1つは、別途開発中のパーソナル逐次型推論マシンとローカルエリアネットワークを介して疎に結合いたしまして、核言語第0版で記述するソフトウェアを開発する為のツールを提供することとございます。後者に関しましては、Prolog をベースランゲージとする核言語第0版についてルール部分の実行PSIが受けもち、大量のファクト部分の処理をこのDeltaが受けもつということで全体として知識情報処理システムの為のソフト開発用支援ツールを構成する、ということになっております。

第2のDeltaを用いたKBM構築のためのデータ収集に関しましてはLANの結合とは別にPSIを密に結合した構成を考えております。又、4台のリレーショナルエンジンをDeltaの中に取り込んでおりまして、これを用いた並列処理の実験等を考えております。

図14は、ホストからのDeltaへのクエリーの流れを示しております。PSIではユーザのクエリーを受けつけまして、コマンド処理プログラムがまずDeltaへの問い合わせプランを作成致しまして、これに基づきDeltaコマンド系列を生成致しまして、パケット形式にし、LANインターフェースを介してDeltaへ問い合わせを行ないます。Deltaはこれを受理し、コマンドの内容を解析致しまして、データベースから所要のデータを検索して再びパケットのかたちでPSIに検策の結果を報告する、という流れになっております。DeltaとPSIとの間のインターフェースはDelta側が関係代数レ

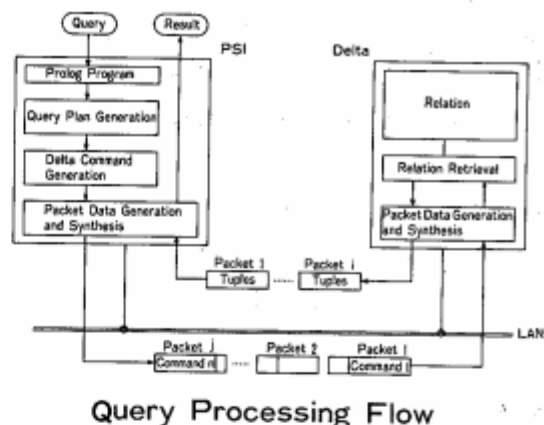


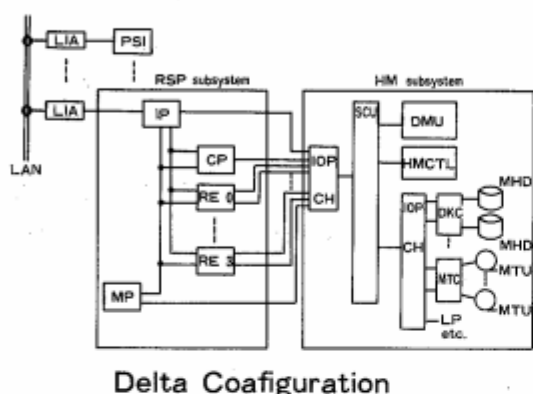
図 14

ベルのインターフェースを提供しております。今までいくつかのクエリーの流れを解析した結果によりますと、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は、次のような特徴があると考えられております。第一はリレーション内のアトリビュートの数、Prologプログラムでは、引数の数とございますが、これはわりあいと少ないということが言えると思います。例えば、先程のデータで示しましたとおり、5ないし10程度とございます。第2に、問い合わせの内容というものは、先程申し上げましたコマンド処理プログラムによりまして、関係演算コマンドに変換されますが、この中でもジョインやユニオン等の比較的負荷の重いものが、多いということとございます。第3に、リレーション内のアトリビュートへどのようなアクセスがかかるかということとございますが、今までPrologプログラムを解析しました結果、各々のアトリビュートに対しまして、負荷はほぼ均等にかかってくる。ですから、どのアトリビュートからアクセスがかけられるかということは、事前には察知できない、どのアトリビュートからも平等なたちでアクセスされることが分かりました。以上の三つの点が非常に特徴的なこととございまして、これらの特性を踏えてマシンアーキテクチャを設定する必要がありソフトウェアシミュレータはPrologで書かれて



ります。

図15はDeltaのシステムの構成を示しております。Deltaは2つのサブシステムから構成されておりまして、全体の制御、監視や演算処理を行なうRSPサブシステムと、関係データベースの格納、検索、変更を担当するHMサブシステムとから構成されております。このDeltaのアーキテクチャ上の特徴はいくつかございますが、例えば先程申

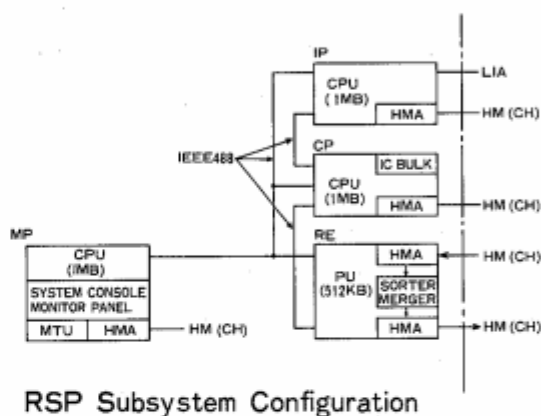


Delta Configuration

図 15

し上げましたように、第一は、関係代数型のコマンドインターフェースを実現しているということとございまして、第2は、この図にありますとおり、機能分散型のマルチプロセッサ構成になっているということとございまして。即ち、5種類、8台のプロセッサから構成されております。5種と申しますのは、コントロールプロセッサ、インターフェースプロセッサ、メンテナンスプロセッサ、リレイショナルエンジン、それからHMコントローラという5種のプロセッサで、エンジン部分を最終的には4台まで増設致しますので5種8台のプロセッサから構成されております。第3に、関係代数専用エンジンを具備した事でありまして、これはHMと2チャンネルのストリームデータ転送のインターフェースで結合されておまして、各々のチャンネルのストリーム転送は、1秒間に3メガバイトの能力を持っております。次にデータベース格納部に階層構造メモリを採用した

ということが、このDeltaの特徴でございまして、可動ヘッドディスクとデータベースメモリユニットの2種類の記憶階層からなっております。データベースメモリユニットの電源障害時の不揮発性を保証するためにDeltaシステム全体にわたりまして、無停電電源装置を使用して瞬断に備えております。それから階層構造メモリ内に格納されるデータが後程申し上げますようにアトリビュートベース内部スキーマ構成をとっているということがDeltaの特徴でございまして。



RSP Subsystem Configuration

図 16

図16はRSPサブシステムの構成でございまして、先程申し上げましたとおり、RSPサブシステムは、コントロールプロセッサ、インターフェースプロセッサ、メンテナンスプロセッサ及びリレイショナルエンジンの各プロセッサから構成されております。CP, IP, MPの各ユニットとも、基本部分はひとつのCPUと512KBあるいは1MBのメモリから構成されておまして、CPではメモリ容量拡張の為に15MBのICバルクメモリを備えております。関係データベースエンジンはパイプラインの2ウェイマージ/ソート方式を実現する為、12段のソート、セルと関係代数演算処理用マージャで構成されております。最大4台のリレイショナルエンジンがコントロールプロセッサで制御されております。

図17はHMサブシステムの構成でございまして。

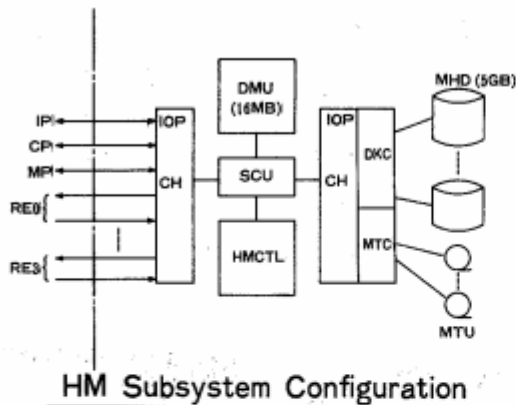
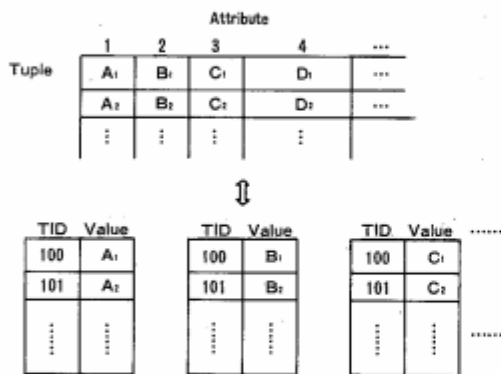


図 17

HMサブシステムはRSPサブシステムの指示に従いまして、Deltaシステムで取り扱うデータの格納領域への効率のよい書き込み、読み出し管理を行なうものでございます。HMハードウェアは、最終構成では128MBの記憶容量を持つ、先程申し上げました不揮発性化した高速DMUと、最大容量が20GBの磁気ディスク装置を備えております。RSPサブシステムとの物理インターフェースはチャンネルで結合されておりまして、チャンネルのストリームインターフェースは先程申し上げましたとおり3MB/Sでございます。

図18は、Deltaの内部スキーマを示しておりま



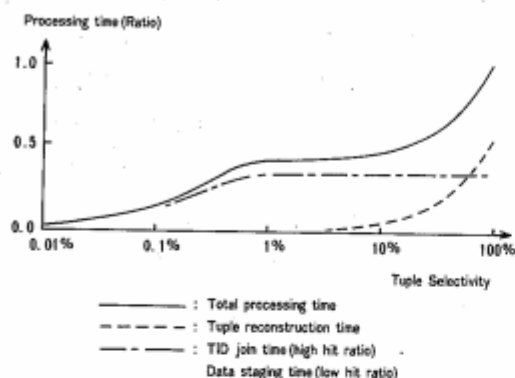
Attribute-based Schema

図 18

す。データベースマシンの性能に大きく影響致します内部スキーマ方式には、従来提案されておりますように、タプル型の格納方式とアトリビュート型の格納方式がございますけれども、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方向に分割致しまして、アトリビュートごとに格納するアトリビュート型格納方式を採用致しました。この理由は3つございまして。第1はPrologをはじめとする論理型プログラミング環境下のDeltaへのアクセスパターンは先程申し上げましたとおり、どのアトリビュートも平等にアクセスする、という特性があります。よって、どのアトリビュートについても平等な管理対象とする必要がある訳でございます。第2は、問い合わせに対しまして、必要なアトリビュートのみをアクセスすればよいこと、これはタプル型の格納方式の場合は不要なアトリビュートまでメモリに読み出してくるというそのオーバーヘッドをきらったわけでございます。第3に我々が採用致しましたリレーショナルエンジンのマージ・ソートアルゴリズムで実現した処理方式、処理効率、及び性能上には、アトリビュート型の内部スキーマの方が適していると判断致しました。以上3つの理由から内部スキーマに関しましてはアトリビュート型の格納方式を採用しております。もちろんアトリビュート型格納方式をとりました結果、いくつかの欠点が指摘されております。第1は出力時にタプルを再構成する、あるいは入力時にアトリビュート方向に分割するという、このオーバーヘッドが必要であるということが第1の欠点でございます。第2は、図に示しましたようにタプルIDを各アトリビュートに付しまして、タプルを識別するというをやらなければいけないわけでございます。当然メモリ領域の効率がおちることになります。この2点が我々の採用しましたアトリビュート型の格納方式の欠点でございます。

さて、現在Deltaは、ハードウェア的にはRSPサブシステム、HMサブシステムがつながりま

したが、ソフトウェア的にそれを現在結合中のございまして、具体的なデータに基づいた性能を今日御紹介するわけにはまいりませんが、私共が予想しております性能の一端を図19に御紹介させていただきます。この机上検討の結果の予測は、リ



Estimated Delta Performance Characteristics

図 19

レーションはまず、タプルあたり、10アトリビュートで1万個のタプルから構成されるという仮定に基づいております。又、各アトリビュートの長さは10バイトとしております。この仮定のもとでセレクトの要求がきた時に、いったいどの程度の性能が得られるかということを経験的に予測したものでございます。ここに示しましたとおり、トータルの処理時間は条件アトリビュートによる適合タプルIDのセレクト及びジョイン処理、出力のためのタプルリコンストラクションの時間の和でございますけれども、第一に、タプルIDで所要のアトリビュートを選択するセレクトの操作はほとんど一定でございます。タプルIDジョインは条件対応アトリビュートによって選択されましたタプルIDをもとに、ジョイン操作を行なう時間でございます。これが割合と長くかかっていると考えられます。又、当然ながらタプル再構成の時間は、選択対象のタプル数が多くなれば、これに比例して増加致します。このグラフで明らかなように、タプルのセレクトビティがだいたい10%以下の領域ではほとんど一定の時間

にレスポンスタイムをおさえることが可能であると考えております。今後各種のコマンド、あるいはテストプログラムを走行、実測致しまして、データの収集を行ない、別の機会に実測のデータに基づいた結果を御報告させていただきたい、と考えております。



図 20

図20は本年5月にハードウェアが結合致しましたDeltaの写真でございます。11月12日から開催されますICOTのオープンハウスでこれを用いまして簡単なデモを行ないたいと考えておりますので、是非おこしいただきたいと存じます。

以上がハードウェアアーキテクチャ関連の研究開発の現状でございますが、前期終了までには今から申し上げますように、いくつかの検討すべき項目を残してございます。

#### Research Items on PIM (Initial Stage)

- (1) More detailed software simulation will be performed for the four parallel inference mechanisms.
- (2) Parallel programs will be investigated to evaluate the four mechanisms.
- (3) Various experiments will be conducted on the experimental machines, each consisting of 8 to 16 modules, to validate each mechanism.

図 21

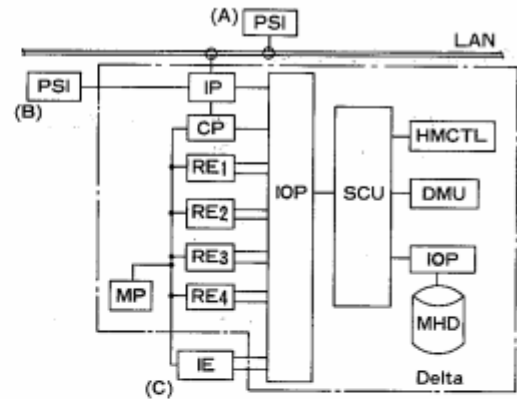
並列マシンに関しましては、図21にありますように、第1には4つの方式に関しまして、より詳細なソフトウェアシミュレーションを行なうこと、第2に、評価のための並列プログラムを開発いたしまして、この並列プログラムについて解析を行なうこと、第3に実験機による各種の実測と評価を行なうこと等が残された問題点でございます。中期に於きましては、これらの前期の研究開発をふまえて、並列推論マシンの構成要素の詳細なアーキテクチャを実際に設計し、各モジュールを試作してみたいと考えております。また、並列推論マシンの全体アーキテクチャ、特に並列推論マシン用のネットワーク、及び専用ソフトウェアの構成を検討いたしまして100台規模の並列マシンについて開発してみたいと考えております。

知識ベースマシンに関しましては、現在関係データベースマシンを開発中でございますが、図22に示す通り3つの問題点がまだ残されていると考えております。ひとつはDeltaハードウェア全体のシステムを実現するということと、ソフトウェアについての統合を完了させまして、ソフトウェアの機能拡張を行なわねばならない事でございます。第2は、LAN経由でPSIを統合した環境で、実際にPrologを走行させてみて実験データを収集致し、Deltaの有効性の評価をしたいと考えております。尚、最大4台のリレーショナルエ

### Research Items on KBM (Initial Stage)

- (1) The entire system of the Delta hardware will be built and functional enhancements will be added to the Delta software.
- (2) Real data will be collected in operational environment consisting of multiple PSIs linked via LAN and the validity of architecture will be checked.
- (3) Algorithms for parallel processing of queries streams will be experimented using up to four relational database engines.

図 22



Tool for Research on KBM Architectures

図 23

ンジンをもとにいたしましたストリームの多重処理アルゴリズムについても、開発検討をしたいと考えております。Prologマシンと関係データベースマシンとの結合、これに関しましては、先程申しあげましたように、LANとDeltaとをネットワークで疎に結合した構成を実現しつつありますけれども、さらに図23に示すように、さらにふたつの結合形体をつくろうと考えております。(A)が今申し上げましたPSIとDeltaのLAN結合による疎結合でございます。(B)ではPSIとDeltaとを密に結合致しまして、インタラクティブなファクトとルール間の処理を行ない、この時の効果の評価をやってみようと考えております。(C)のインタフェースではインフェレンスエンジンをDelta内に取り込み、これによりまして、知識ベースマシンに向けて知識ベースの処理というものがいったいDeltaの中でどう構成されるべきであるか、ということについての検討をしたいと考えております。これら3結合方式の結果をもとに致しまして、知識ベースマシンに関しましては、中期では第2次試作を行なう予定でございます。さらに、分散知識ベースの制御メカニズムにつきましても、研究を展開したいと考えております。

最後になりましたけれども、以上御報告致しましたICOTに於けるハードウェア、及びアーキ

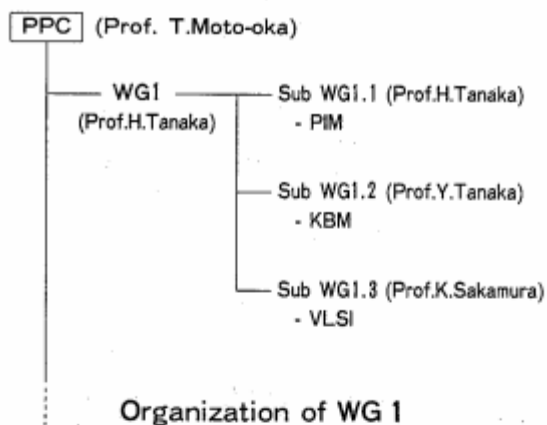


図 24

テクチャの研究開発は、ICOTの研究センターのメンバ、及び担当のメーカーの方々のみならず、図24に示しましたように、元岡先生を委員長といたしますプロジェクト推進委員会の配下のワーキンググループ1、およびこの中の3つのサブワーキンググループのメンバの方々にも御協力いただいております。この機会をおかり致しまして、皆様方のご協力・ご指導に感謝の意を表したいと思います。どうもありがとうございました。