

TM-0132

昭和 60 年度電子通信学会
情報・システム部門全国大会発表論文集

ICOT ならびに関連機関

September, 1985

©1985, ICOT

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

単一化による知識ベース検索

Retrieval By Unification on Knowledge Base

横田 治夫

Haruo YOKOTA

安部 公朗

Masaki ABE

森田 幸伯

Yukihiko MORITA

伊藤 英則

Hidegori ITOH

(財) 新世代コンピュータ技術開発機構

Institute for New Generation Computer Technology

1. はじめに

データベース・システムが発達したのは、単に格納すべきデータの量が増大したためだけでなく、データのモデルが確立され、各種のアプリケーションからデータの集合を共有することが可能になったことに大きく依存している。知識ベース・システムにおいても、単なるデータよりもさらに複雑な構造を持つ知識の集合を多数の利用形態に提供することが重要になると思われる。つまり、知識を統一的に取り扱うために、知識の供給者と利用者との間に知識のモデルを設定することが必要となる。

現在、知識のモデルとしていくつかの知識表現が提案されているが、大量の知識を効率的に共用することに重点を置いて考えられた知識表現は少ない。我々は、関係データベース・システムを発展させた形の知識ベース・システムとして、関係モデル上で知識の操作を行う方法を提案する。

このことにより、関係データベースで培ったデータ共有のノウハウが知識ベースにも利用可能になるとと思われる。

2. 関係型知識ベース

関係モデルはデータの集まりを集合として数学的に取り扱うのに非常に適している。ただ従来の関係データベースでは、その集合の各要素（データ）を、定数のみからなる有限領域上で定義された関係の個々の要素を表現するものとして扱っている。従って、関係データベースの各要素に対する操作としては定数どうしのイコリティ・チェックのみが行われ、各要素は構造物として認識されることはない。

我々は、関係操作の対象を項（term）、つまり変数を含む一種の構造物に拡張し、基本操作として項間の単一化（unification）を導入する。項は構造を持った無限領域上で定義された関係を表現することを可能にするので、知識の持つ複雑な構造を取り扱うことができる。

形式的には従来の関係データベースの操作の対象は、

$$d \in R \subset D_1 \times D_2 \times \dots \times D_n \quad (D_i \text{ is a set of constants})$$

であり、 R のことを関係と呼んでいた。関係型知識ベースの操作の対象は、

$$k \in T \subset K_1 \times K_2 \times \dots \times K_n \quad (K_i \text{ is a set of terms})$$

となる。ここで、 T のことを項関係と呼ぶことにする。関係に名前が付けられたように、項関係にも名前を付けることができる。

関係データベースに対する演算としては、関係代数と関係論理が基本的な演算として採用されている。我々は、関係型知識ベースに対する演算として、実際の処理に直接落とし易いという観点から、関係代数を拡張することを考える。ただし、関係型知識モデルとしては項の外延、すなわち項の基礎例（ground instance）の集合とし、関係型知識ベース処理はそのモデルを表現する項関係上での操作とみなす。

関係代数における基本的な演算としては、射影（projection）、結合（join）、制約（restriction）がある。ここで、射影演算は集合の各要素に対して処理を行わないので、演算対象が定数から項に変わっても演算自体は変化しない。結合演算と制約演算については、要素間のイコリティ・チェックを単一化に拡張する。すなわち、単一化結合（unification-join）と単一化制約（unification-restriction）という2つの新しい演算を導入する。

別の観点から見ると、これらの演算処理は項関係で表現した知識に対する単一化を用いた検索と見なすことができる。我々は、これらの処理を総称してRBU（Retrieval By Unification）と名付けた。

3. RBUによる演繹

次に、関係型知識ベース操作の一例として、ホーン節を2属性の項関係に格納し、単一化制約及び単一化結合を繰返すことにより、演繹が行えることを示す。

まず、ホーン節を帰結部と条件部に分けそれぞれを一つの属性とし、一つの節を一つのタプルに対応させて格納する。条件部と帰結部の間で単一化を行うことにより演繹が進められるように、条件部と帰結部それぞれを2進木で表現し、含まれる述語を各葉に対応させ、一番最後の葉を変数とする。

次のようなホーン節があった場合、

$$P \vee Q_1 \vee Q_2 \vee \dots \vee Q_n$$

これを図1のような二つの木で表現する（図中のXは変数）

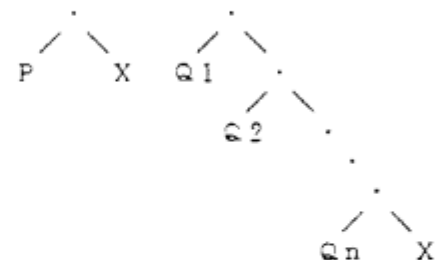


図1 関係型知識ベースにおけるホーン節の表現

例を用いて演繹の過程を見る。知識として、親は祖先であり親の祖先もまた祖先であるという規則と、各親子関係の事実が、図2のような形態でk b 1という名前の項関係に格納されているものとする。ここでは、2進木の表現としてDEC10-Prologのリスト表現を用いている。

この関係型知識ベースに太郎の祖先を問い合わせる場合、まずk b 1の第一属性に対し、

$$[\text{ancestor}(\text{taro}, A)]$$

という項で単一化制約をかけ、temp1（図3）という

kb1	
[ancestor(X,Y) S]	[parent(X,Y) S]
[ancestor(X,Y) S]	[parent(X,Z), ancestor(Z,Y) S]
[parent(taro, ichiro) S]	S
[parent(ichiro, shigeki) S]	S
...	...

図2 項関係の例

temp1	
[ancestor(taro, A)]	[parent(taro, A)]
[ancestor(taro, A)]	[parent(taro, Z), ancestor(Z, A)]

図3 単一化制約の結果

temp2	
[ancestor(taro, ichiro)]	[]
[ancestor(taro, A)]	[ancestor(ichiro, A)]

図4 単一化結合と射影の結果

新しい項関係を作る。

次に、この temp1 の第二属性と kb1 の第一属性との間で単一化結合を行ない、temp1 の第一属性と kb1 の第二属性が残るように射影をかけて、temp2 (図4) という項関係を作る。

さらに、この単一化結合と射影の組合せを新しく出来た項関係と kb1 との間で繰り返すことにより、与えられた頂上節 (goal) に対する全解を求めることができる。

ここで、途中で作られる temp1, temp2 等の項関係の第二属性は、入力導出法 (input resolution) において、頂上節あるいは前段階の導出形 (resolvent) と各入力節との間に得られる導出形である。また、第一属性は、頂上節中の変数に対する代入を残すためのものである。第二属性の内容が空リストになった場合は、空節が導かれたことを示しており、第一属性で示される変数の代入により反駁が得られたことを示している。

例では、temp2 の一番目のタプルは、空節を導出形としていることから一つの解であり、太郎の祖先の一人は一郎であることを示している。また、二番目のタプルの第二属性は一郎の祖先を導く副頂上節であり、一郎の祖先は太郎の祖先であることを示している。

途中で作られる項関係のあるタプルの第二属性が空リストであったり、リストの先頭の述語と単一化可能な項が元の項関係の第一属性に存在しない場合には、その枝に対する演繹はそこで停止する。途中で作られる項関係の全てのタプルの演繹処理の続行が不可能になったところで、全解が求まったことになり、処理を終了する。これは、次に作る項関係のタプルがないことで示される。現在の Prolog 処理系で停止するようホーン節集合については、全解を求めて停止する。

本方式は探索木のあるレベルの全ての導出形を求め、ため、横優先 (breadth-first) 探索で全解探索を行っていることになる。2進木 (リスト) の最後の変数に、各枝に対するスタック処理を実現していると見

ることができる。あるいは、述語単位の置換えを行っていることから、ストリング・リダクションを実現していると見ることもできる。

4. 共用性と汎用性

一階述語論理が大きな知識表現能力を持つことは、良く知られている。一階述語論理の部分集合であるホーン節が関係型データベースで取り扱えることは、前節で示した。

関係データベースの類推から、関係型知識ベースでは、複数のユーザからホーン節の一つの集合を同時にアクセスすることができ、ホーン節の事実 (fact) のみならず規則 (rule) の量が巨大になった場合にも有効であることが分かる。

ホーン節以外の知識表現としては、意味ネット、プロダクション・システム、フレーム等が提案されている。また、関係データベースもある種の知識表現と見ることができる。

関係データベースについては、格納対象を定数とするだけで、そのまま関係型知識ベースで取り扱うことができる。意味ネットについては、DCKR [1] としてホーン節の演繹に置き換えることができる。プロダクション・システムやフレーム等については、構造を持った表表現に対する検索処理とみなすことができ、関係型知識ベースと親和性が高い。

我々は、RBUによる関係型知識ベースの検索システムの上に、知識コンパイラ [2] を構築し各種の知識表現を取り扱うことを検討している。

5. おわりに

関係データベースを拡張した関係型知識ベースを提案し、拡張した関係代数の繰り返しでホーン節の演繹が可能であることを示した。関係型知識ベースは、ホーン節だけでなく、他の知識表現を格納するのに適しており、汎用性が高い。また、一つの知識を複数のユーザで共有することができる。

現在我々は、本方式をマシンで実現することを検討している。以前我々は、関係データベース・エンジンと階層構造メモリを基本構成要素とする関係データベース・マシン Delta を開発した。関係データベース・エンジンを単一化機能を提供した関係型知識ベース・エンジンに置き換えることにより、知識ベース・マシンを構成することが考えられる。今後、実現方法のさらに細かい検討を進めていく。

【謝辞】本検討を進めるに当たり、有益なご示唆を頂いた知識ベース・マシンWGのメンバーの方々、および熱心なご討論を頂いた、神、東芝、日立のKBM会議メンバーの方々に感謝します。

【参考文献】

- [1] 小山, 田中: Definite Clause Knowledge Representation, Proc. of the Logic Programming Conference '85, pp95-106.
- [2] 村上, 他: 知識ベースマシン "Delta 2" の知識コンパイラ, 昭和60年度通信学会部門別全国大会.
- [3] 横田, 他: 知識ベースマシン構築のための一考察, 情報27回 (昭和58年後期) 全国大会予稿, 2K-7.
- [4] Yokota, H. et al: Unification in A Knowledge Base Machine, 情報29回 (昭和59年後期) 全国大会予稿, 3F-2.

知識ベースマシン'Delta 2'の 知識コンパイラ

Knowledge Compiler of Knowledge Base Machine
'Delta 2'

村上 昌己 横田 治夫 物井 秀俊 伊藤 英則
Masaki Murakami Haruo Yokota Hidetoshi Monoi Hidenori Itoh
新世代コンピュータ技術開発機構

Institute for New Generation Computer Technology (ICOT)

1. はじめに

第五世代コンピュータ計画中期4年間で開発を計画している知識ベースマシンDelta 2は、複数ユーザの多様なアプリケーション間での、知識資源の共有化を実現することを意図している。そこでDelta 2の基本的な設計思想においては、特定の知識表現やアプリケーションのもとでの高いパフォーマンスを求めるよりも、既存、または近い将来考えられうる様々な知識情報処理に対する拡張性のあるシステムを意識している。

以上より、Delta 2のソフトウェアシステムは、次の3つの機能をサポートするものと考えられる。

- (1) 知識表現に依存しない格納形式。
- (2) マルチユーザをサポートするLAN接続等による、バックエンドOS。
- (3) 多様なアプリケーションとの論理的なインタフェースを支援する知識ベースマネジメント。

本稿では、(3)の機能をサポートする部分として開発を計画している知識コンパイラIKC(Intelligent Knowledge Compiler)についてその基本的考え方と構想を述べる。

2. 目的

共有型の知識ベースシステムに対する要求を、データベースシステムの延長として考えると、次のような点があげられる。

i) 知識独立性: データベースシステムにおけるデータ独立性の概念に相当するのが知識独立性である。知識を複数のアプリケーションプログラムで共用するためには、各々のアプリケーションプログラムが知識ベースシステム内での知識表現の構造や形式に依存しないで作成できなければならない。逆に言うならば、知識ベースに格納される知識は、データベースにおける概念スキーマと同様に、知識そのものの論理的な構造に基づいた形式で格納されるべきである。

ii) 知識ベース検索の柔軟性: 従来のデータベースに求められる検索の柔軟性に加えて、知識ソースに属しまたは除に含意される情報を、各々のアプリケーションに都合のよい形で引き出す操作に対応できることが求められる。例えば、知識がHorn節の形で表現されていた場合を考える。ある述語 $foo(X, Y)$ の定義があったとき、従来の演繹型データベースのように、 foo を満たす X と Y の組が求めたい場合の他、データベースにおけるaggregate functionのように X と Y の組の数が求めたい場合、また $foo(X, Y)$ の定義自身が欲しい場合等がある。また、同じ知識ベースを利用する複数のアプリケーションが、それぞれロジック、フレーム、意味ネットワーク等の異なった知識表現が用いられた

場合、それぞれの形式で検索が可能であることが望ましい。

このように同じ知識ベースに対する多様な検索要求を解釈し、知識を加工処理する機能のうち、いくつかのアプリケーションで共有可能なもの、また大量の知識を一度に扱う必要があるもの等は、知識ベースマシン内でサポートした方が有利である。

iii) 知識の一貫性・無矛盾性の維持: この問題に関してはデータベースの完全性制約(integrity constraint)の維持の問題に加えて、知識の意味的な制約の維持の問題がある。即ち、知識ベースが新たにインストールされた段階で、欠落や矛盾を含んではならず、また知識ベースの更新は意味的に「合法的な」形で行われなければならない。これらの問題は知識獲得の問題として研究が進められてきたが(國藤, 他)、共有型の知識ベースにおいては、さらに先の知識独立の問題と絡めて考えられなければならない。

知識コンパイラIKCは、以上のような問題について、論理型プログラミングにおけるメタ・プログラミングの枠組みで統一的に扱うインテリジェントな知識ベース管理システムをめざす。

3. 基本的方針

知識ベースマシンDelta 2は関係型知識ベースを直接取り扱うことが可能なマシンであり、知識を項関係の形で格納し、RBUによって操作する(横田, 他)。IKCはRBUとホストの間にはいるものとして位置づけられる(伊藤, 他)。ここではホストとして推論マシン(PSI)を想定していること、また知識コンパイラにおいてメタ・プログラミングの技法を多く活用する関係から、原則としてIKCの扱う項関係はロジックによって記述された知識を表現しているものを想定している。即ち項関係はfactとruleの集まりであるtheoryを表している。項関係の名前は即ちtheoryの名前であり、IKCにとって知識ベースはtheoryの集まりである。前章で述べた共有型知識ベースに対する要求i)~iii)を扱うため、IKCでは次のような概念を用意している。

① Virtual Theory: 関係データベースシステムにおいては、データベースに対する操作は、関係名を引数に含む演算として処理された。知識ベースに対する操作は、同様に考えると、theory名を引数に含むようなメタ述語の呼出と考えられる。例えばtheory t で述語 $foo(X, Y)$ が定義されているとき、 (X, Y) を求める操作は、

$demo(t, foo(X, Y), Result)$

というメタ述語の呼出となる。

前章で述べた知識独立性の実現を考えると、関係データベースにおけるviewの概念と同様に、ユーザは、自らが用

いるアプリケーションにとって自然であるようにtheoryを編集して用いることが可能であることが望ましい。即ち項関係で記述されたtheoryの集まりである知識ベースを概念スキーマとし、ユーザはその上にな仮想的なtheory (virtual theory)を定義し、ユーザのアプリケーションは、知識ベースに対する操作をvirtual theoryを引数に含むようなメタ述語の呼出として扱えるといった形態が考えられる。IKCでは、このようなvirtual theoryの定義をメタ知識として知識ベースに格納し、アプリケーションからvirtual theoryの呼出が生じたとき、それらのメタ知識をデモンとして活用する機能をサポートすることを考えている。

②キューリ解析：知識ベースに対する操作要求は多様であり、ロジックで表された知識ベースに対する検索だけをとっても先に述べたように、演繹型データベースのように、述語名を与えてそれを満たすインスタンスを求める場合、また述語の定義そのものが欲しい場合などがある。前者の場合は、demoまたはset-ofのような、述語を評価する類のメタ述語の呼出と考えられ、後者の場合は、マクロ化したclause述語のように述語を引用する類のメタ述語の呼出と考えられる。また、theoryはDelta2の内部ではリレーションであり、各factやrule、即ちtupleの順序は考慮されていない。そこで、例えばあるtheoryを、Prologのようなdepth-fast, linear なサーチに適するように順序を付ける等の加工処理をサポートすることも考えられる。

IKCではこれらの解析・処理の手続きを、知識の操作に関するメタ知識として格納し、ユーザの処理要求をこれらのメタ知識を検索しながら解析することを考えている。

また知識ベースに対する操作の定義を知識ベース内に格納したことによって、ユーザが、知識ベースに対する操作を自由に拡張できる可能性を考慮している。

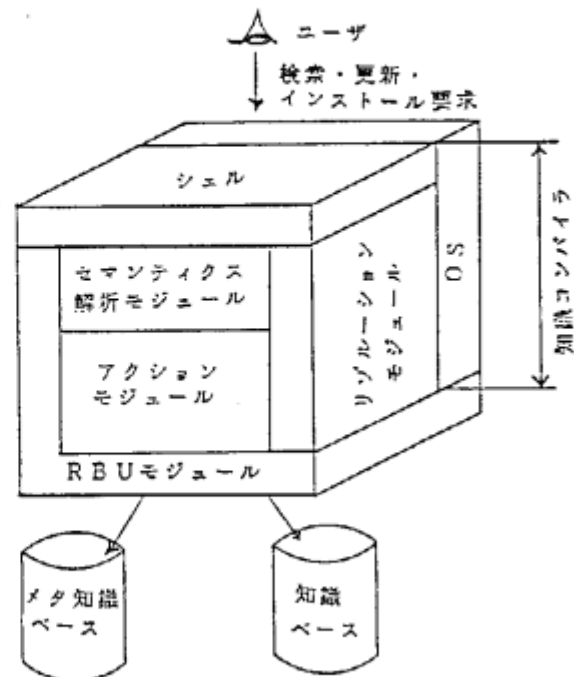
以上のようなキューリの解析処理によって、知識ベースに対する要求ii)に対応すると同時に、先のvirtual theoryのサポートと併せて、i)の知識独立性を実現することを考えている。

④メタ知識による無矛盾性維持：知識ベースシステムにおける無矛盾性の維持は、データベースシステムの無矛盾性チェックが、言わばプログラムのコンパイル時に検出可能なエラーのチェックのレベルに留まっていたのに対し、知識を表現する記法のコンストラクトの意味をある程度考慮した上で考えられるべきである。即ち、無矛盾性の維持が特に問題となるのは、知識を新たにインストール及び更新した場合であるが、このとき、知識表現のコンストラクトの意味を定義するメタ知識、特に知識に関するコンストレイントを与える類の知識を使うことになる。

IKCでは、このような無矛盾性の維持に関しては、インストール・更新時に、ホストからのインストール・更新の要求を解析し、コンストレイントを参照するメタ述語を呼出すか、あるいはデモンの形で記述されたコンストレイントについてはそれらを直接呼出すよう、インストール・更新時のインターフェースを制限することによって実現することを考えている。

4. 知識コンパイラの構成

以下に、IKCを中心としたモジュール構成を図示する。



各モジュールの働きは以下の通りである。

- (1)シェル：知識コンパイラに対するユーザからの操作要求について、検索・インストール・追加のいずれに類するか大まかに解析し、各モジュールの呼出を制御する。
- (2)セマンティクス解析モジュール：知識コンパイラに対する入力に先に述べたようにメタ述語の呼出で行われる。このメタ述語の定義及び引数に現れる述語、theoryを定義しているメタ知識の収められたtheoryを呼び出すモジュールである。
- (3)アクションモジュール：セマンティクス解析モジュールで用意されたメタ知識を参照し、virtual theoryの上の操作を実際に項関係で格納されているtheoryの上での操作に変換すること、及び用意されたメタ知識のうちデモンとして実行されるべきものを実行する。
- (4)リゾリューションモジュール：項関係で記述されたtheoryの上で、リゾリューションを実行する。
- (5)RBUモジュール：RBUの演算を実行する。知識コンパイラの各モジュールはRBUの演算の呼出で記述されている。

5. おわりに

Delta2の知識コンパイラについて、基本的考えかたを述べた。今後、Delta2の開発、知識ベースのモデルの形式化と並行し、パッケージとして実現を進める予定である。
謝辞：有益な御示唆をいただいた知識ベースマシンWGの皆様、及び熱心な御討論を頂いた、沖、東芝、日立のKB M会議メンバーの皆様へ感謝します。

(参考文献)伊藤、他「知識ベースマシン・Delta2」
「開発の背景と方針」、横田、他「単一化による知識ベース検索」以上2件昭和60年度通信学会部門別全国大会。
国藤、他、ICOT研究論文、TR-130

会話音声理解についての検討

Consideration on Conversational Speech Understanding

小松昭男 大平栄二

Akio KOMATSU Eiji OOHIRA

日立製作所・中央研究所

Central Research laboratory, HITACHI, Ltd.

1. はじめに

計算機の高度利用が進むにともない、人間と計算機との自然な知的対話機能が重要となる。そして、人間にとってきわめて自然な情報交換手段である音声の利用が必須となる。本報告において、会話音声理解についての基本的な検討を行い、制限の無い自然な会話音声を理解するための技術開発の足掛かりとする。

2. 会話音声理解の概要

自然言語理解の研究は、理解の対象としている文章の種類と、その表現手段とによって、図1に示すように分類できる。会話音声理解は、会話における自然な言語、すなわち、音声によって表現され伝達される会話文の文意を理解するもので、自然言語処理の基本的な過程であると言える。

3. 会話音声理解に固有な問題点の検討

会話音声理解の対象とする会話文は、記述文とは異なった表現方式であり、それに起因した固有の問題を含んでいる。これらは次のようにまとめられる。

(1) 文分割の問題 会話文では、複数の文が続けて入力されるほか、思考を組み立てながら発話されるため、言い間違いなどによる不整文が少なくない。さらには、構文上はあまり意味の無い「えー」などのような感動詞が多い。しかも、それらは、句読点によって区切られてはいない。このため、理解の単位である文や文節への分割が困難である。

これに対し、発声内容に対して自然的・合理的な情報である、音声固有の韻律情報の利用が有効と考えられる。

(2) 曖昧文の問題 会話文は曖昧であり、文

法的に整っていない場合が多く、しかも、文脈に大きく依存する。また、会話音声文字コード列に変換（連続音韻認識）することは困難である。特に、語尾や文末において発声が不明瞭になる場合が多い。このため、格文法の中心的役割をなす用言が文末近くに現われる場合には問題になる。

これに対し、文意伝達の核となる概念語などの単語（重要単語）を検出し、その組み合わせで文意を推定する方式が有効と考えられる。

4. 会話音声理解方式の検討

会話文と記述文とでは、表層レベルの表現形態が異なる。従って、自然言語処理の過程からみれば、形態素解析レベルでの処理方式は異なるが、意味解析レベルの処理は、基本的に同じである。構文解析レベルでは、会話文がタスクに依存しているため、汎用的な高精度構文解析方式をベースにして、タスク対応への展開が必要となる。

これとともに、会話での情報伝達において、音声のもつ音韻情報と韻律情報とが果たす機能などを考慮し、以下の3つの処理部よりなる会話音声理解方式を提案する（図2）。

(1) 文構造推定部 - 韻律情報を処理し会話文の分割などを行う。(2) 重要単語推定部 - 音韻情報を処理し会話文中の重要単語の検出を行う。(3) 文意推定部 - 音声・言語・タスクの各情報を統合的に処理した構文解析・意味解析を行う。

4.1 文分割方法の検討

会話文を意味的なまとまり（文法的に整った文や文節とは限らない）に分割する方法について検討する。会話音声における韻律情報は、(1)

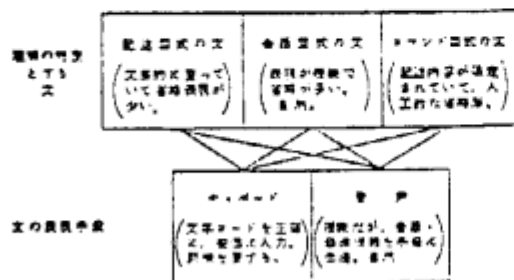


図1 自然言語理解システムの分類

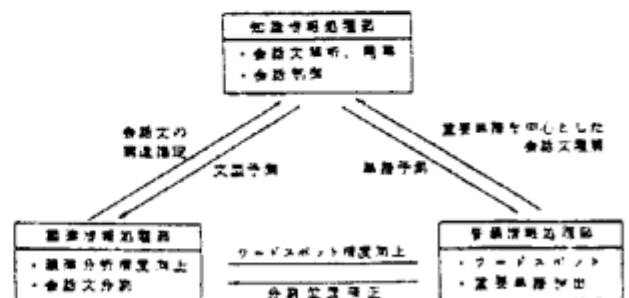


図2 会話音声理解方式の概念図

(1)基本周波数、(2)音声パワー、(3)発声テンポ、として観測される。これらは独立ではなく、相互に関連しながら変化し、発声内容や意図などを反映している。また、息づぎや構文からの拘束条件により変化する。

このような韻律情報のもつ性質を利用して、会話文の文分割が可能である。具体的には、(1)無音区間長、(2)無音区間前後の基本周波数差、(3)無音区間前の最低周波数値、を考慮して、会話文の文分割を行う。図3に分割候補点での韻律情報の形状を示した。

4.2 重要単語検出方法の検討

会話文の核となる重要単語は、最終的には、構文・意味解析レベルの処理によって検出されるものである。しかし、韻律情報が発声内容や意図に関係し、その概形をよく表現していることから、韻律情報を利用した重要単語位置の推定が可能である。たとえば、会話文中のある単語が、意味的に重要な場合には、発声テンポは自ずからゆっくりで丁寧になり、音声パワーは大きく、基本周波数は高くなる(幸いなことに、そのような部分の認識精度は一般にはよい)。また、特に意識的な強調の場合、付属する格助詞に特有のピッチパターンが観測される。

これらの現象は一般的なものであり、重要単語検出のための利用価値は高い。

5. サンプル会話音声の解析

会話音声理解について、より具体的に検討を進めるため、実際の会話環境下での会話実験を行い、会話文サンプルを解析する。

タスクとして、電話交換業務を対象とし、交換手役と利用者役との間での自然な会話の実験を行い、以下の会話文サンプルを得た(図4に会話音声例を示す)。

交1: こちらは...でございます

利1: おはようございます

交2: おはようございます

利2: ...と申します

交3: はい

利3: 内線1234の...をお願いしたいんですが

交4: はい、お待ちくださいませ

これは、ごく普通の会話であるが、その会話音声の解析・検討の結果、以下のことが分かった。

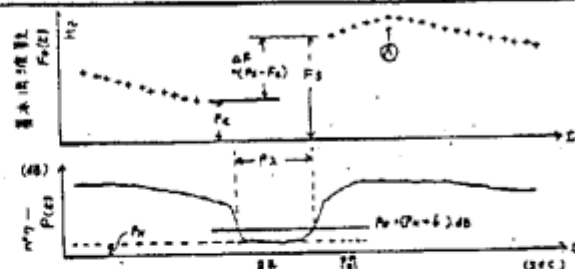


図3 分割点の韻律情報の形状

(1)会話文は文末まで聞いて理解されてはいない。利用者の挨拶が終わる前に交換手の挨拶が始まっているし、交換手の挨拶が終わる前に利用者が発声し始めている。このようなオーバーラップは随所に見られる。

(2)重要単語は丁寧に発声されている。「内線」という単語は、相手に情報を伝達するための核となる単語であり、丁寧に発声されている。発声速度は7-8音節/秒である。

(3)新しい情報の無い部分の発声は雑である。

「...をお願いしたいんですが」という部分は、定型句であり、双方にとって重要な情報を含んではいない。発声は丁寧ではなく、相手も注意して聞いてはいない。発声速度は約15音節/秒である(音韻認識はきわめて困難)。

これらの解析結果は、いずれも、これまで述べた検討結果とよく整合している。

6. まとめ

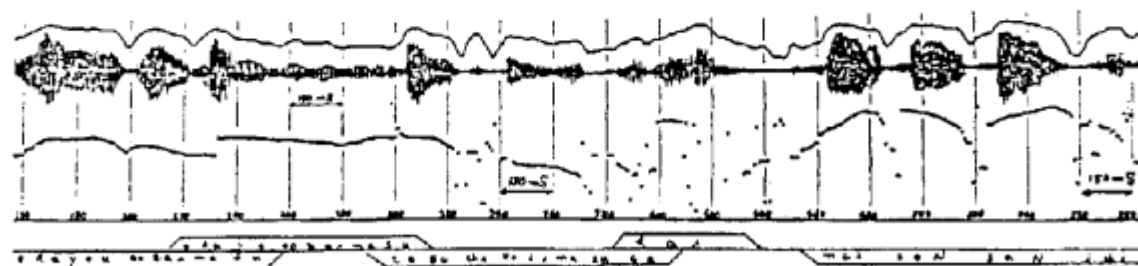
会話音声理解についての基本的な事項についての検討を行い、会話音声理解方式の概念レベルでの提案を行った。また、実際の会話環境下の会話サンプルの解析より、会話音声理解についての検討内容の妥当性を認めた。なお、本研究は、第5世代コンピュータ・プロジェクトの一環として、ICOTからの委託により行ったものである。

参考文献

- 1) 田中他, 自然言語処理技術と言語理論, 電経研調報205号(1981).
- 2) 新美, 音声認識, 共立出版(1981).
- 3) 鹿野, 会話音声自動認識システムに関する研究, 名大博士論文(1981).
- 4) 藤崎, 日本語単語アクセントの基本周波数パターンとその生成機構のモデル, 音学誌, 27-9(1981).
- 5) W. A. Lea, A Prosodically Guided Speech Understanding Strategy, IEEE Trans. Vol. ASSP-23, NO. 1(1975)

図4
会話音声例

スケール:
縦: 10dB
横: 100ms



マルチポート・ページメモリの構成方法

Construction of Multiport Page-Memory

横田 治夫
Haruo YOKOTA物井 秀俊
Hidetoshi MONOI森田 幸伯
Yukihito MORITA伊藤 英則
Hidenori ITOH

(財) 新世代コンピュータ技術開発機構

Institute for New Generation Computer Technology

1. 背景

ディスク等二次記憶装置を用いて大容量のデータを処理するようなシステムでは、CPUと主メモリ間のボトルネック（フォンノイマン・ボトルネック）だけでなく、二次記憶と主メモリ間のボトルネックが大きな問題となる。

このデータ処理ボトルネックを解消するために、複数の二次記憶装置および複数のデータ処理装置から同時にアクセス可能なマルチポート・メモリを大容量データ処理システムに利用する方法が提案されている[1, 2]。特に田中が提案しているマルチポート・メモリ[2]は、アクセスをページ単位とすることにより、各ポートから待ち時間なしで同時にアクセスすることができる。

我々は、このマルチポート・ページメモリの実現法を検討し、経済的かつ実用的な構成方法を考案した。

2. 基本概念 [2]

マルチポート・ページメモリは、基本的には図1のように、複数のポートと複数のメモリ・バンクと、そのポートとメモリ・バンクを接続するスイッチング・ネットワークからなる。

複数のポートから同一のページを同時にアクセス可能にするため、論理的なページを全てのメモリ・バンクにまたがるように配置する(図1)。また、スイッチング・ネットワークも、どのメモリ・バンクにもアクセス競合が生じないように、ある時点でのポートとメモリ・バンクの接続関係を固定にし、クロックで切り換えるようにする。

アクセス単位をページとし、ページの途中から読み書き可能とすると、このマルチポート・ページメモリは全てのポートから待ち時間なしで同一ページをアクセスすることができる。

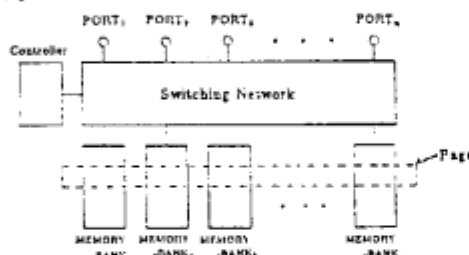


図1 マルチポート・ページメモリの概念図

3. 実現方法

マルチポート・ページメモリを実現しようとした場合、特にポート数が多い場合、ポートとメモリ・バンク間のバス幅が実装上大きな問題となってくる。

そこで、バス幅をなるべく狭くするように、アドレスをデータ・バスを用いて転送する。ただ、1データごとにアドレスを転送しては非効率なので、ページ単位もしくはそれ以上の単位でアドレスを指定し、ページ内のアドレスについてはメモリ・バンク側で計算する。

ここで、マルチポート・ページメモリの場合1ページが

全メモリ・バンク間にまたがるため、スイッチング・ネットワークの切り替えと同期させてメモリ・バンク間でアドレスを転送する必要がある。このため、各メモリ・バンクにアドレス転送用のシフトレジスタを設け、隣のメモリ・バンクと接続する。また、スイッチング・ネットワークの切り替えも、アドレスのシフトと同方向に一つずつ隣に切り替わるような接続をできるようにする。

さらに、このメモリ・バンク間のアドレス転送は、ポートからのメモリ・アクセスと同時に進めたい。そこで、ページ・アドレス分のラッチを用意し、ページ・アドレスをシフトしている間アクセス用のアドレスを固定させておく。また、最低限、メモリ・バンク間でアドレスを転送している間は、各ポートとメモリ・バンクの接続は切り替えずにアクセスするように1メモリ・バンク当たりのアクセス単位を設定する。

スイッチング・ネットワークは、ビットスライス化して構成する方法が考えられる。各ビット分については、ルータを用いて多段ネットワークを構成する方法も考えられるが、切り替え方式がローテートで、ポートの数が4~64程度の場合には、PLAを用いて構成するほうが容易である。切り替えは、ポートの数を n とすると n 進カウンタを用いればよい。

各ポートについては、待ち時間なしでアクセスするためには、ページ内のアドレスをスイッチング・ネットワーク接続関係から計算する必要がある。単なるローテートのネットワークの場合には、切り替え用カウンタの値を使うことにより加算器のみで容易に求めることができる。

4. まとめ

マルチポート・ページメモリにおいて、データ・バスを用いてアドレスを転送するようにした場合の構成法について報告した。このように構成することにより、実装上の難点が少なくなり、経済的に実現することができる。このマルチポート・ページメモリを関係データベース・マシン、知識ベース・マシン等に利用することにより、データ処理ボトルネックの解消に効果があると思われる。

【謝辞】本検討を進めるに当たり、有益なご示唆を頂いた知識ベース・マシンWGのメンバーの方々、および熱心なご議論を頂いた、沖、東芝、日立のKBM会議メンバーの方々に感謝します。

【参考文献】

- [1] Boral, H., et al.: Implementation of the Database Machine DIRECT, IEEE Trans. on SE, Vol. SE-8, No. 6, pp533-543
- [2] Tanaka, Y.: A Multiport Page-Memory Architecture and A Multiport Disk-Cache System, New Generation Computing, Vol. 2, pp241-260.

IKC: Intelligent Knowledge Compiler

RBV: Retrieval By Unification

KB: Knowledge Base

これらの各ユニットの機能と特徴を概説する。

(1) KB

- ・ term で記述された知識を格納する。
- ・ IKC が各機能を実行するために予め用意してある知識 (メタ知識) を含む。

(2) RBV

- ・ KB 内の知識を unification より検索する。
- ・ KB 内の大量の知識を効率的に unify させるために、知識をストリーム状にしてパイプライン処理させるメカニズムを持つ。
- ・ 検索した知識に対して推論処理を行う。

(3) IKC

- ・ IKC はホストと RBV の間に設けて、個別利用者に対して KB 内知識および検索された知識の最適化を行なう。さらに個別利用者に対してキュアリ処理の最適化を行う。
- ・ 無矛盾性の保持および View 定義を行う。
- ・ 以上の処理に際しては KB 内のメタ知識を用いる。

5. まとめ

以上知識ベースマシンの一般化の立場からその基本的事項について述べた。

Delta 2 の開発に当たっては知識表現は、ホストでの知識表現レベルと統一するために、図 2 に示す一階述推論の一部である definite clause とする [3]。このことにより、ホストとの処理分担率の向上に柔軟性が期待できるようになる [4] [5] [8]。

k \ l	0	1	>1
0		type 1 constant variable	type 5
1	type 2	type 4	type 6
>1	type 3		

一階述推論式

$$P_1 \wedge P_2 \wedge \dots \wedge P_k \rightarrow Q_1 \vee Q_2 \vee \dots \vee Q_l$$

----- : Delta 1 知識表現レベル

----- : Delta 2 知識表現レベル

図 2 知識表現レベル

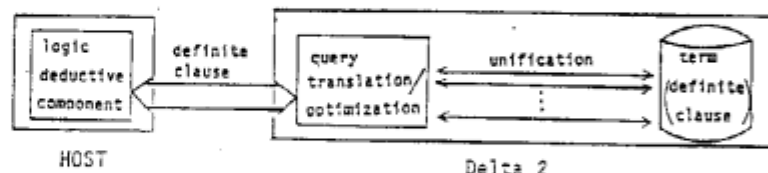
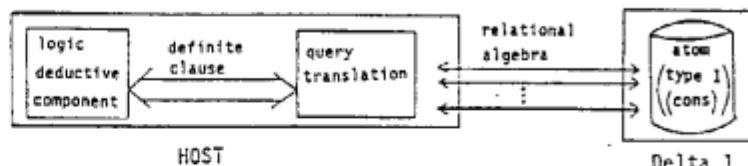


図 3 ホスト-Delta インタフェース

ホスト-Delta 2 間インタフェースにおいては definite clause タイプの言語の使用が可能となりそのインタフェースの単純化が図れる。また、ホスト側ではキュアリのトランスレーション処理が不要となる。以上よりホストの負荷軽減が図れる。

また、知識の検索に RBV 機能を設け、格納知識に対して知識ベース内でその推論処理を可能としたことにより、知識情報処理システムトータルとして処理効率を上げることが可能となる。

今後は IC、RBV 機能の詳細化、KB 内知識格納構造の明確化をすすめる。また、データをストリーム状にしてパイプライン処理するエンジンのメカニズムおよび、2 次記憶装置とエンジン間の入出力の効率化を検討する。

【謝辞】 本検討を進めるに当り、有益な示唆を頂いた知識ベースマシンワーキンググループメンバーの方々、および熱心なご討論を頂いた、沖、東芝、日立の KBM 会議メンバーの方々に感謝する。

【参考文献】

- [1] 村上: アーキテクチャとハードウェアシステム: 第五世代コンピュータ国際会議 FGCS '84 会誌録, pp.74-86(1985)
- [2] K. Murakami, et al.: Research on parallel Machine Architecture for Fifth-Generation Computer System, Computer, pp.76-92, June (1985)
- [3] H. Gallaire, et al.: A Deductive Approach, Computer Surveys, Vol. 16, No. 2, June 1984
- [4] 小山、田中: Definite Clause Knowledge Representation, conf. of the L.P.C.'85, pp.95-106
- [5] 横田、他: 知識ベースマシン構築のための一考察, 情報第 27 回全国大会予稿, pp.679-680
- [6] F. Hayes-Roth, et al.: Building Expert Systems, Addison-Wesley(1983)
- [7] H. Gallaire: Logic Data Bases vs Deductive Data Base, Logic programming Workshop 1983, pp.608-622
- [8] 大須賀: データベース応用技術の将来展望, 京都大学大型計算機センター第 16 回研究セミナー報告(1985), pp.97-118

つぎに、Delta 2 とホストとのインタフェースに、注目してその特徴を述べる (図 3 参照 [7])。

知識ベースマシン “Delta 2”

— 開発の背景と方針 —

Knowledge Base Machine “Delta 2”

— Background and Principle of Development —

伊藤 英剛 横田 治夫 村上 昌己 大堀 雅博 角田 健男
Hidenori Itoh Haruo Yokota Masaki Murakami Masahiro Ohba Takeo Kakuta

新世代コンピュータ技術開発機構

Institute for New Generation computer Technology (ICOT)

1. はじめに

第5世代コンピュータシステム研究開発プロジェクトでは、知識情報処理に向けたコンピュータの開発のため、推論・知識ベース機構の実現を図っている。

その基本要素の一部として、前期3年間(S57～S59)で関係データベースマシン Delta 1 を開発してきた[1], [2]。この開発を通して、関係で表現されたデータの格納とその操作に関する技術、ノウハウを蓄積してきた。

中期4年間(S60～S63)では、格納する知識表現の拡張、知識管理のインテリゼント化、ホストインタフェースの高度化、格納されている知識を大量に必要とする推論処理は知識ベース内でも処理可能としてトータルシステムの効率化を図り、知識ベースマシン(Delta 2)のプロトタイプを開発する。本論文では、Delta 2 開発の基本的考え方、方針等について述べる。

2. 背景と狙い

近い将来知識は、より顕著に次のような特徴を有するようになると考えられている。

- ① 共有化 — 蓄積された知識は種々の利用目的で、かつ種々の利用者に使われ得るために共有化がすすむ。
- ② 大量化 — 知識の応用範囲が広がるとともに新しい知識が蓄積され大量となる。
- ③ 高級化 — より高度な問題を解くための知識や専門的な知識が必要となる。

以上の動向に対する知識ベースマシンDelta 2 の開発の狙いを以下に述べる。

①知識独立(マルチユーザ、マルチパーパス)

知識の共有化を図るため、知識ベース内に蓄積される知識を利用者から独立させてマルチユーザ、マルチパーパス向き知識ベースマシンとする。

②知識処理分担(バックエンドタイプ)

格納知識の大量化、知識独立化に応え易いために、知識ベースマシンをバックエンドタイプとする。また、知識情報処理システムトータルとして処理の効率化を図るために、大量に格納されている知識を必要とする推論処理については、知識ベースマシン側でその実行を分担する。

③知識表現レベルの統一(知識検索の高度化)

知識情報処理システム全体(ホスト、バックエンド)で知識表現言語を同一レベルとすることにより、ホストの知識検索の高度化とホスト、知識ベースマシン間インタフェースの単純化を図る。

3. 基本方針

(1) 知識表現

知識の表現は、一階述語論理、セマンティックネットワーク、フレーム等の手法が考えられている。Delta 2 ではこれらの知識表現を term (項) で表現して知識ベースに格納する。

(2) 知識検索

知識を term で表現し格納することから、知識の検索処理にunification 機能が必要となる。このために知識ベースの基本的検索機能としてRBU (retrieval by unification) の概念を導入し、その処理機能をつける。

(3) 知識管理

知識ベースマシン内の知識は、知識の独立化が図られていることより、利用者に対して知識ベース内知識を個別対応に最適化、および、利用者のキュアリに対してキュアリ処理の解析のために知識コンパイルなどの知識に関するインテリゼントな管理機能をつける。[6]

4. 基本アーキテクチャ

知識ベースマシンDelta 2 の基本アーキテクチャを図1に示す。

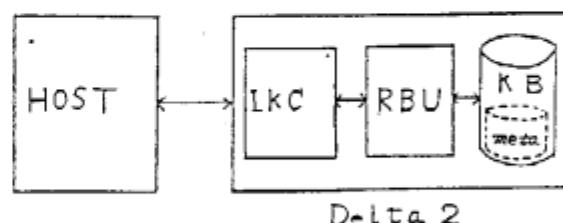


図1 基本アーキテクチャ概念図

知識ベースマシンDelta 2 をつぎの3つの基本ユニットから構成する。

関係データベースマシン Delta 1 における 階層構造メモリの制御方式とその評価 HIERARCHICAL MEMORY CONTROL METHOD AND ITS EVALUATION IN THE RELATIONAL-DATABASE MACHINE Delta 1

*長谷川 忠弘

Tadahiro HASEGAWA

* (株)日立製作所

Hitachi LTD.

*石川 博道

Hiromichi ISHIKAWA

** (財) 新世代コンピュータ技術開発機構

Institute for New Generation Computer Technology (ICOT)

**伊藤 英則

Hidenori ITOH

1. はじめに

ICOTにおいて、関係データベースマシンDelta 1を開発した。Delta 1は、関係演算処理部(RSP)と階層構造メモリ部(HM)から構成されている。

HMは、Delta 1におけるデータ格納庫であり、RSPからのデータ処理要求に対し、高速かつ大量のデータ処理が円滑に行なわれるよう設計されている(図1、Delta 1の構成参照)。

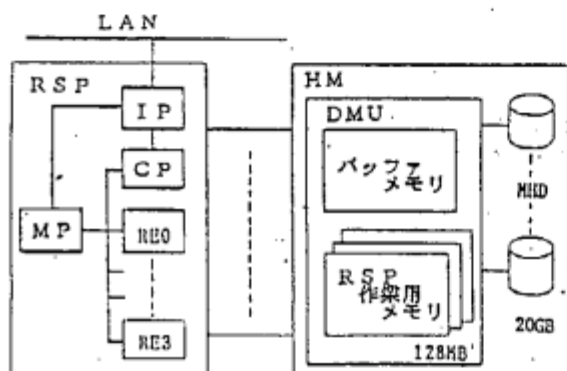
本稿では、HMの制御方式とその評価について述べる。

3. 評価

HMの評価を行うために、Delta 1のサブシステムの一つであるREと接続しデータ収集を行った。この結果について述べる。

3.1 実ページ数と性能

図2は4台のREを接続し、同時にデータを処理したときの経過時間である。実ページ数がある値以上になると経過時間が逆に増加している。これは、1つのバッファidに対し割当てる実ページ数が多いと1回にリプレースする量も増加し、リプレースメント中の非恒制御による待ち時間が増加するためである。



IP:Interface Processor CP:Control Processor

MP:Maintenance Processor

RE:Relational Database Engine

DMU:Database Memory Unit MHD:Moving Head Disk

図1. Delta 1 の構成

2. HMの制御方式

HMは、可動ヘッドディスク(MHD)とDMUの二階層のメモリで構成している。またDMUは、バッファメモリとRSPの作業用メモリから成る。バッファメモリは、512B、4,096Bそして32,768Bの各ページ長毎にサブプールとして管理し、リプレースメントアルゴリズムとしてはLRU(Least Recently Used)を使用している。作業用メモリは、32ビットのバッファidによって識別される複数の仮想記憶である。各仮想記憶は、32KBのページ枠から構成され、最大256MBまで割当てることができる。1つの仮想記憶に割当てた実ページ枠数は、システムパラメタで与える。割当てられた実ページ枠数を超えて要求すると直前までのページ枠をMHD上のページとリプレースする。

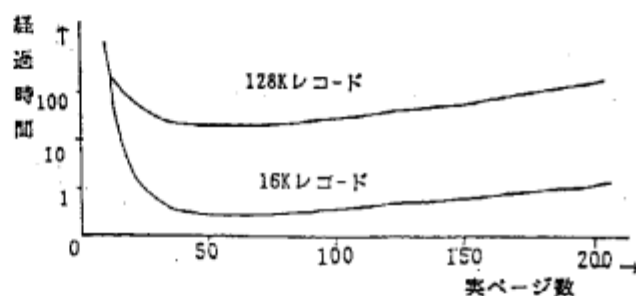


図2. ページ数と処理時間

3.2 アクセス特性

HM内トレースにより、作業用メモリは順アクセスであることが明確になった。またMHDへのアクセスも順アクセス主体でありLRUアルゴリズムの欠点である『順アクセスによる更新処理は、MHDのヘッド位置が乱れる』現象が発生しやすい。さらにLRUアルゴリズムは、バッファサーチを行っているため、バッファ面数に比例してサーチオーバーヘッドが増加している。

4. おわりに

HMの評価を行うことにより、Delta 1におけるデータのアクセス特性を明確にすることができた。

Delta 1の開発および評価を行うにあたり御協力頂いた(株)東芝の各位に感謝致します。

参考文献

- (1) 宮崎 他、「データベースマシンの機能設計について」、第29回 情報全国大会(1984)
- (2) N. Miyazaki, "An Overview of Relational Database Machine Delta", 情報シンポジウム, 「Advanced Database」 (1984)

関係データベースマシンDelta1における コントロールプロセッサの制御方式とその評価 CONTROL METHOD OF THE CONTROL PROCESSOR AND ITS EVALUATION

IN THE RELATIONAL DATABASE MACHINE Delta1

伊藤 文英⁺ 星野 康夫⁺ 角田 健男⁺⁺ 伊藤 英則⁺⁺

Fumihide ITOH Yasuo HOSHINO Takeo KAKITA Hidenori ITOH

⁺(株)東芝 ⁺⁺(財)新世代コンピュータ技術開発機構

Toshiba Corporation Institute for New Generation Computer Technology

1. はじめに

関係データベースマシンDelta1は通産省第5世代コンピュータプロジェクトの一環として開発された。[1] Delta1はホストマシンである推論マシンと接続され、外部データベースを提供する。Delta1の構成を図1に示す。

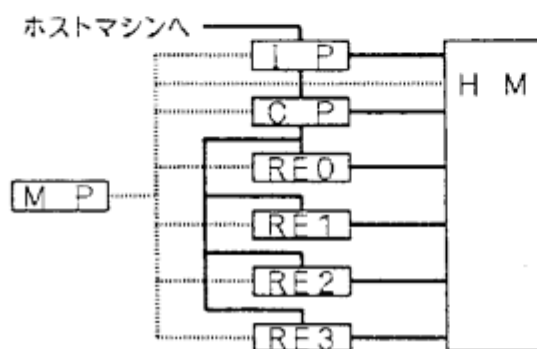


図1 Delta1の構成

インタフェースプロセッサ（IP）はホストとの通信を制御し、ホストからのデータベース問合せ（コマンド木）をコントロールプロセッサ（CP）に送る。CPはコマンド木を解析し、階層構造メモリ（HM）及び関係データベースエンジン（RE）に命令（サブコマンド）を発行して、演算処理を制御する。HMは大量のデータを格納し、サブコマンドに従って演算に必要なデータを準備する。REはHMからデータを入力し、サブコマンドで指定された演算を行って、結果をHMに出力する。演算結果はIP経由でホストに転送される。このほか、メンテナンスプロセッサ（MP）がシステムの保守および運用を制御する。

2. CPの制御方式

CPでの演算処理は次の3つのフェーズに大別される。

①コマンド木構文解析

コマンド木の構文を解析し、アクセスされる関係にロックをかける。

②サブコマンド列生成

関係の定義情報を参照しながら、HM及びREへのサブコマンド列を生成する。

③サブコマンド列実行

生成されたサブコマンド列を逐次的にHM及びREに発行する。サブコマンド列実行処理には1台のREが割り当てられ、最大4つのサブコマンド列が並列に実行される。ただし、CPでは1つのホストからのコマンド木は1度に1個しか処理しないので、並列実行されるサブコマンド列は全て異なるホストに対するものである。

3. 評価

Delta1に対する代表的な問合せの1つとして Selection を取上げ、データ量の異なる次の2つのケースの処理時間についての測定結果を検討する。

①ケース1

256レコードから1レコードを選ぶ。

②ケース2

16384レコードから256レコードを選ぶ。

3. 1 各処理フェーズの所用時間比

CPにおけるコマンド木処理の各フェーズの所用時間の、ケース1の全体を100とした比率を表1に示す。データ量の差によるHM及びREにおけるサブコマンド処理時間の差は、コマンド木処理全体に影響を与えるほどではない。

表1

	ケース1	ケース2
解析	18	18
生成	17	17
実行	65	84
全体	100	119

3. 2 サブコマンド並列実行の効果予測

Delta1では関係を属性方向のデータの集まりとして格納し、各データにはタプル方向のつながりを示すためのタプル識別番号（TID）がつけられる。従って、関係に対する演算は原則として属性に対する演算として実行され、また、各属性のデータに対してTIDによるSortや Selection が行われる。[2]

ケース1、ケース2とも結果レコードは4つの属性からなり、各属性に対する Selection 及び Sort が発生する。これらを4台のREを同時に使用して並列に実行した場合の、HM及びREにおけるサブコマンド処理時間の和の、ケース1の並列実行を100とした比率を表2に示す。

表2

	ケース1	ケース2
逐次実行	100	434
並列実行	41	187
並列/逐次	41%	43%

4. おわりに

Delta1に対する代表的な問合せに対して、CPの性能測定及び処理方式の評価を行った。

Delta1の試作及び評価に関し御協力いただいた（株）日立製作所の方々に感謝いたします。

参考文献 [1] 角田、柴山、横田他「RDBM Delta (I) ~ (III)」情報処理学会第26回全国大会講演論文集、4F-6~8, 1983

[2] 伊藤他「関係データベースマシン“Delta”における関係代数演算の最適化」情報処理学会第30回全国大会講演論文集、10-1, 1985

関係データベース・エンジンの性能評価

Performance Evaluation of the Relational Database Engine

安部公明 伊藤英則
Masaaki Abe Hidenori Itoh
(財)新世代コンピュータ技術開発機構
Institute for New Generation Computer Technology

岩田和秀 柴山茂樹
Kazuhide Iwata Shigeki Shibayama
(株)東芝
Toshiba Corp.

酒井浩
Hiroshi Sakai

1. はじめに

関係データベース・エンジン(以下、REと略称する)は、Deltaの主要な演算ユニットであり、ソート処理を主体とした関係代数演算を専用ハードウェアで、またハードウェアでサポートされていない演算と全体の制御をソフトウェアで行うことにより、関係データベースに対する処理を効率良く実行できるように設計されている。

本稿では、REの性能評価について報告する。

2. REのハードウェア構成

関係データベースに対する処理を効率良く実行するためには、ソート処理やJoin、Selection等の関係代数演算を高速に行う必要がある。このため、REでは、ソート処理と関係代数演算処理を連動して、入力データに同期したパイプライン処理を行う専用ハードウェアを開発した。

REの構成を図1に示す。HMアダプタは、HMとREのインタフェース制御を行い、入出力独立に同時動作可能である。INアライナは入力データのフォーマット変換を行うハードウェア、ソータは2-way merge sortアルゴリズムによりソート処理を行うハードウェア、マージャは関係代数演算処理専用のハードウェアである。マイクロプロセッサは、全体の制御とエンジン・コアでは処理できない演算(算術演算や集約演算など)を行う。

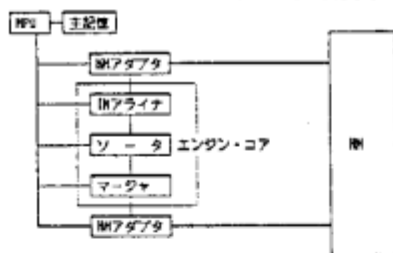


図1 REの構成

3. REのデータ処理方式の概要

REで処理される演算は、マイクロプロセッサが上記各モジュールを制御することにより実行される。REがJoinのような2つのリレーションを扱う演算を実行するならば、次に示す処理を行う。HM上に格納されている2つのリレーションのうち片方のリレーション(IN₁)をHMアダプタを介して受取り、INアライナでモディファイし、ソータでソートし、最後にマージャのパッファに格納する。次に、もう一方のリレーション(IN₂)も同様な処理を行った後マージャの別のパッファに格納する。IN₂をソートしている間にマージャは既にパッファに格納されているIN₁の各レコードと比較を行い、結果を生成する。演算結果リレーション(OUT)は、HMアダプタを介して再びHMに送られる。

ただし、エンジン・コアで処理できない演算については、自身のマイクロプロセッサで高速処理するため、CPから送られてくるコマンドをコンパイルしてマイクロプロセッサが直接実行可能なオブジェクトコードを生成し(コンパイル方式を採用)、演算対象データを主記憶上に転送し、処理を行う。

4. REの性能評価

4.1 性能評価方法

- (1) システム構成 HMと1台のREを用いる。
- (2) 演算対象データ 演算対象リレーションは、REで作成、HMに転送後、HMのRSP作業用メモリ上に格納される。レコードの長さは、すべて16バイトとする。
- (3) 処理時間の測定方法 REのコマンド解釈からレスポンスの作成までの時間間隔をREの計時モジュールを用いて測定する。

4.2 結果

エンジン・コアによる処理時間は理論的に見積ることができるので、理論値と実測値を比較して期待どおりの性能が得られているかどうかを確認する。図2、3にJoin演算の結果を記す。紙面の関係でJoin演算しか記せないが、当日にはREの各種演算の性能評価の検討を行う予定である。

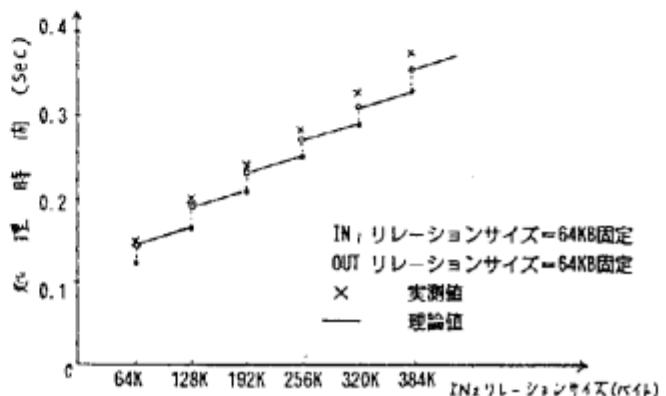


図2 JOINの性能評価(1)

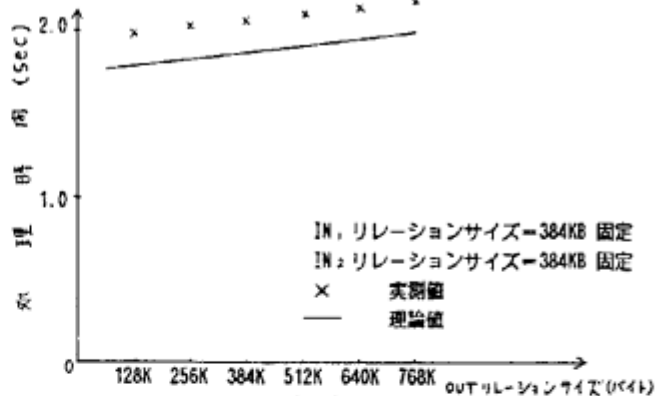


図3 JOINの性能評価(2)

5. 考察とまとめ

実測値が理論値よりも大きくなるが、これはコマンドの解釈やハードウェアの起動、HMでの処理に要する時間などのソフトウェアのオーバーヘッドを含むからである。図2の理論値は階段状に上昇しているが、これはマージャの中の2つのリレーションを格納するパッファに限りがあり(64KB×2)これを越える場合には、64KBごとに処理を行うためである。図より、REは関係代数演算を期待どおりの時間(データ量nとして、処理時間O(n))で処理することがわかり、REの専用ハードウェアの処理能力の高さが十分に発揮されていることが確認された。また、ソフトウェアで処理する演算もコンパイル方式を採用することにより、マイクロプロセッサが処理することを考慮すると十分に高速に処理していることが確認された。今後、4台あるREの並列処理の性能評価を行う予定である。

[参考文献]

- (1) 安部、酒井他、「関係データベースエンジンの開発(その1~6)」情報処理学会第29回全国大会予稿
- (2) Sakai, H., et al: Design and Implementation of the Relational Database Engine, Proceedings of the International Conference on Fifth Generation Computer Systems, 1984