

大規模知識ベースマシン実験機の開発(1)

-開発の背景と方針-

3B-1

伊藤 英則 物井 秀俊 森田 幸伯 横田 治夫*
 柴山 茂樹⁺ 酒井 浩⁺ 石村 多喜二⁺ 岩田 和秀⁺
 (財)新世代コンピュータ技術開発機構 東芝⁺ 富士通^{*}

1. はじめに

第5世代コンピュータ・プロジェクトではハードウェアシステムとして推論マシンと知識ベースマシンを開発している。

横田等は、項で表現した知識の知識ベースモデルとして関係型知識ベースモデル⁽⁶⁾を提案した。これを大規模知識ベースマシンに適用し、その実験機を開発する。関係型知識ベースモデルの基本演算は項の集合間の単一化関係演算である。この処理をパイプライン化し複数台の専用処理プロセッサで並列実行して効率化することが課題である。なお、専用処理プロセッサの処理単位をページとする。処理対象を論理構造体(term)，基本演算を単一化関係演算、処理単位をページ、処理を並列化すること等の特徴に注目して以下を述べる。

2. 関係型知識ベース並列処理モデル

いま、 $R(A, B)$ を集合A, Bの関係演算とする。A, Bが大きい場合は一台の処理プロセッサが一回の処理でできる粒度に分解し、以下の各々についての関係演算を実行する。

$$R(a_1, b_1, b_2, \dots, b_m) \text{ --- (1-1)}$$

$$R(a_2, b_1, b_2, \dots, b_m) \text{ --- (1-2)}$$

...

$$R(a_i, b_1, b_2, \dots, b_m) \text{ --- (1-l)}$$

ここに、 $A = (a_1, a_2, \dots, a_l)$, $B = (b_1, b_2, \dots, b_m)$

この、(1-1) ~ (1-l)を複数の処理プロセッサでいかに並列実行させるかが課題である。

2.1 単一メモリアクセス-並列演算処理モデル (1A-nP)

前期で開発した処理プロセッサ4台を持ったDELTAはこのモデルであった。これは複数の処理プロセッサをもつが同一集合Aに同時にアクセスを許さない。複数のプロセッサが存在していても、(1-1), (1-2), (1-3) ...の順に実行される。すなわち、異なる集合間の関係演算に対しては並列に処理できるが、同一集合間の関係

演算の並列処理はできない。なお、Aを a_1, a_2, \dots, a_n に分割し、かつBを処理プロセッサ分コピーすれば並列処理が可能であるが、コピーオーバーヘッドと分割オーバーヘッドができる⁽³⁾。

2.2 多重メモリアクセス-並列演算処理モデル (nA-nP)

このモデルはA, Bともに複数の処理プロセッサからのアクセスが許可される。すなわち、 $R(a_i, b_1), R(a_i, b_2), \dots, R(a_i, b_m), i=1 \sim m$ を複数台の処理プロセッサで同時に実行させることができ(図1参照)、上述のオーバーヘッドなしで処理効率をn倍に向上できる。

3. 関係型知識ベースマシンアーキテクチャ

nA-nPモデルのマシンアーキテクチャはMPIS⁽⁴⁾(多重プロセッサ、間接検索)を基本として実現する。メモリ機構は多重アクセス、ページ単位の処理のためにMPPM(Multi-Port Page Memory)⁽¹⁾⁽²⁾を適用する(図2参照)。MPPMはn個の知識格納ユニット(バンク)を、また各バンクは一つのポートを持つ。n個のバンクに物理的にまたがったページはバンク毎の各ポートからアクセスされる。各ポートはプロセッサにスイッチングネットワークを介して接続され、プロセッサとバンクの接続は相対位置関係を守って一定周期で切り替わる。この関係型知識ベースマシンアーキテクチャについて、1.で述べた特徴からみた考察・解析を行う。

4. 処理プロセッサ

処理プロセッサは項をデータストリームで入力してパイプライン処理により関係演算を行なう。なお、処理効率化のためにA, Bを整列後に関係演算を実行するのが一般的である。しかし、複数のプロセッサでの整列処理は処理段数が深くなり後段になるほど並列度が下がる⁽³⁾ため、集合が大きい場合は全体を整列することはプロセッサの使用効率上得策ではない。このため、A, Bを分解したそれぞれ a_i, b_j の単位で整列させ、(1-1), (1-2), (1-3) ...を複数の処理プロセッサで並列処理させる方が得策である。

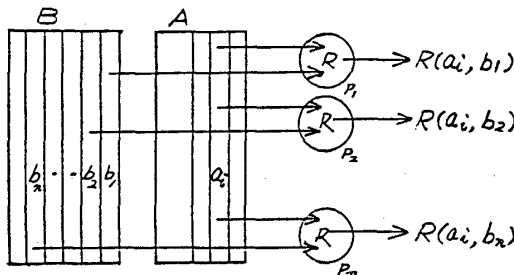


図1 nA-nPモデルの概念図

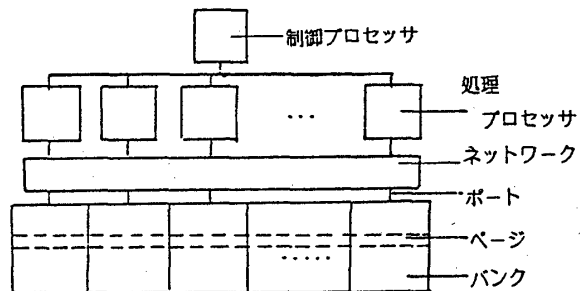


図2 基本構成概念図

ここに a_i, b_j の単位で整列させるためには処理プロセッサの1回の入力容量(後述のbuffer memory)は上述の a_i, b_j より大きいことが必要である。

4.1 等価関係演算

等価関係演算は結合演算(join)に代表される。結合演算の前に整列すれば処理量オーダーは $O(n)$ にできる。整列は $O(n \log n)$ かかるが $\log(n)$ 段の並列化アルゴリズム⁽⁵⁾により $O(n)$ にでき、結合は整列と組み合わせ $O(n)$ にできる。これはDELTAで実証した⁽⁷⁾。

4.2 単一化関係演算

単一化関係演算は単一化結合演算(unification-join)⁽⁶⁾に代表される。単一化結合演算は単一化による項同志の結合に4.1を拡張したものである。処理効率化のため、項の整列化は単一化からみた項の一般性により行なう⁽⁸⁾。また、知識ベース検索に対する処理は一般に全ての解を求め尽くすまで単一化演算を繰り返すパターンが多いと想定される。特に、RBU (retrieval by unification)⁽⁶⁾はこの特徴を持つ。この特徴は検索処理の過程で4.1と比べて大量の中間結果を発生させることが予想できる。なお、知識を定義する変数の多少、論理的構造の複雑度、等による中間結果量・処理量についての評価が必要である。

4.3 処理プロセッサの基本構成

等価関係演算処理プロセッサとしてRE (relational engine)を開発した。単一化関係演算処理プロセッサとしてUE (unification engine)を開発する。UEの基本構成要素⁽⁷⁾⁽⁸⁾は以下とする。

- (1) ソータ (pipe-lined 2-way merge sort⁽⁵⁾) $\times 2$
- (2) バッファメモリ $\times 2$
- (3) ペアジェネレータ (RE のマージャに相当)
- (4) ユニファイヤ (RE のコンパレータに相当)

(1)~(4) 個々の機能検討に加えて、キー属性以外の項を処理プロセッサ内で、どのように流して最後の出力部で合流させるかは処理効率上の課題である。

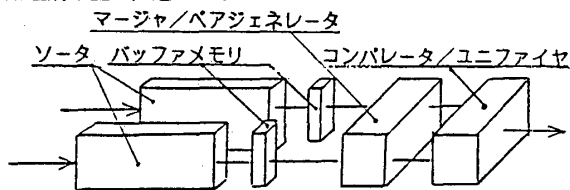


図3 処理プロセッサ構成概念図

5. 並列資源管理/並列実行管理

5.1 コマンド解析

コマンドの解析によりプリミティブ演算と処理フェーズに分解し、各フェーズにおける並列度、処理粒度、処理量の動的解析を行なう。関係単一化演算の場合には、中間結果の処理粒度および処理量の予測解析は困難であることから、実行制御戦略と関連付けて考察する必要がある。

5.2 並列実行管理

処理プロセッサのバッファメモリ内に収まり得るほど一方の集合が小さければ、片方が大きくても効率的処理ができることを実

証した(3)(7)。これより、selection first 等により一方の集合を小さくする演算実行順序制御戦略が必要となる。またユーザ特性に依存するが使用頻度の高い方をプロセッサ内に滞留させ入出力負荷削減等の考慮も必要となる。また、資源割当て制御戦略は資源の負荷状態と演算特性によりどの資源を優先するかについての動的並列制御戦略が必要となる。

5.3 並列資源管理

システムの物理資源としては複数プロセッサ、バッファメモリ容量、ページ数、バンク容量がある。並列実行最適化制御のためにはこれらの負荷状態を監視する必要がある。プロセッサ処理速度、データストリーム速度、ポートのリード/ライト速度、DISK I/Oおよびアクセスタイムからこれらのシステム定数の相関とバランスの解析については並列資源管理/並列実行管理方式の双方から総合的に解析を行なう必要がある。

6. 実験機開発の方針

- ・MPPM/スイッチングネットワーク；このモデルのインフラストラクチャであり、早期に開発する。
- ・処理プロセッサ；ソフトウェアシミュレーションによる方式的解析と同時に、処理プロセッサの実験機の設計を進め、最終的に μP に組込むかまたは μP をリプレースする。
- ・制御プロセッサ；将来は複数の推論マシンにリプレースする。
- ・制御プログラム；並列資源割当て、並列実行制御についてソフトウェアシミュレーションによる解析結果を活かして実現する。

7. おわりに

処理粒度をページ単位とする大規模知識ベースマシンとして複数処理プロセッサの並列制御/多重メモリアクセスモデルの実験機について述べた。このマシンアーキテクチャはポート数の増加に柔軟性を欠くが、評価用ツール類を完備すること等により実際のポート数 $\times n$ 倍の解析法を考察する。

参考文献

- (1) H. Borat, et al. Implementation of the Database Machine DIRECT IEEE Tran. on S.E. Vol. SE-8, NO. 6 Nov. 1982.
- (2) Y. Tanaka, MPDC: Massive Parallel Architecture for Very Large Databases, Inter. Conf. on FGCS. pp. 113-137, Nov. 1984.
- (3) H. Itoh, C. Sakama, et al. Parallel Control Techniques for Dedicated Relational Database Engines, ICOT-TR-182.
- (4) K. Hwang, F. Briggs Computer Architecture & Parallel Processing McGraw-Hill, 1985.
- (5) S. Todd. Algorithm and Hardware for Merge Sort Using Multiple Processors, IBM J.R&D. Vol. No. 5, Sep. 1978.
- (6) H. Yokota, H. Itoh. A Model and an Architecture for a Relational Knowledge Base, 13th Int. Symp. Comp. Archi. 1986.
- (7) S. Shibayama, H. Sakai A Relational Database Machine. 10th Symp. Comp. Archi. June 1983.
- (8) Y. Morita, H. Yokota et al. Retrieval-by-Unification on a Relational Knowledge Base Model, 12th. VLDB. 1986.