

連想メモリを用いた並列推論マシンの検討

4B-6

長沼 次郎 小倉 武 木村 隆

(NTT 電気通信研究所)

1. はじめに

推論等の知識情報処理の高速化を目的とした並列推論マシンの研究が活発化している。実用的な知識情報処理システムを実現するためには、100KLIPS~1MLIPS規模の推論プロセッサを数百台規模で、並列処理する技術の確立が不可欠である。

一方、我々は連想メモリの適用により、並列推論マシンへの拡張を意図とした逐次推論マシン：ASCAの検討を行ってきた⁽¹⁾。ASCAはインタプリタで100KLIPSの高速性を有しており、さらに各種スタック領域に連想メモリを適用し、これを抽象化することにより、マルチプロセッサ型の並列推論マシンへの拡張性を追及している。

本稿では、ASCAを要素プロセッサとするマルチプロセッサ型の並列推論マシンの計算機モデルとその実現手法に関する検討結果を述べる。

2. 計算機モデル

高い並列台数効果を得るために、並列化のためのオーバーヘッドの排除とプロセッサ間の通信量の低減化に主眼をおいた。本並列推論マシンでは、論理型言語の実行におけるゴールのリダクション過程を、複数の逐次推論プロセッサにおいてスタックベースで実行制御する。各プロセッサのスタック領域への連想メモリの適用がこれらの課題を容易にしている。基本的な構成は、複数台のASCAをネットワークで接続したものである。概念図を図1に示す。

(1) 対象とする並列性

AND、引数間並列処理は、変数束縛の一貫性チェック等のオーバーヘッドを考慮して採用しない。各プロセッサ間の処理の独立性、逐次型からの自然な拡張⁽²⁾⁽³⁾を考慮して、OR並列処理を採用した。

(2) 基本処理単位と内部表現

基本処理単位は、ゴールのリダクション過程における選択枝を含むORプロセス(以下、単にORプロセスと呼ぶ)である。個々のORプロセスは、初期ゴールの解を独立に得ることができる。ORプロセスの内部表現は、複数のスタックを用いて表す。すべての構造体データは、Structure-Sharingで表現し、プロセッサ間で共有メモリ等は持たない。

(3) 負荷分散

ORプロセスは、内部でバックトラックできるため、発生するORプロセスの数をプロセッサの台数以内に制限可能である。基本的には、空状態のプロセッサが発生したら、ORプロセスを分割する。その際、自らの中でバックトラックすべきか、他のプロセッサに分割すべきかを、分割点の推論深さを評価して決定する。

(4) 推論深さ空間の拡張

容量限界に達したプロセッサは、空状態のプロセッサを見つけ、それを下位に連結して実行を継続する。これにより、処理可能なプログラム規模が拡大される。

3. 実現手法

(1) 推論の実行

推論の実行は、ASCAにおけるPrologの逐次実行と同様である。空状態のプロセッサの認識は、節の起動時にネットワークからの情報によって行う。

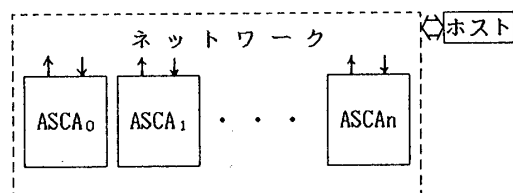


図1. 本並列推論マシンの全体構成

(2) ORプロセスの分割

ORプロセスの分割の概念図を図2に示す。分割点は、連想メモリで構成されたバックトラックスタックの検索によって容易に見出すことができる。分割点と現時点の推論深さをを用いて、分割すべきかどうかを判定する。分割点に付随する述語呼出しスタック、バインド情報スタックの内容が空状態プロセッサに転送される。この時、転送されるゴールに含まれる変数の評価を行い、必要なバインド情報のみを転送し、通信量を軽減する。また、変数評価時の連想メモリの検索を契機として、自らの分割点までの不要なバインド情報のガーベジコレクションを行う。

なお、初期ゴールに対するORプロセスの分割においては、自らのプロセッサ識別子を用いて、各プロセッサが異なるORプロセスを選択する。概念図を図3に示す。また、各プロセッサ内の実行環境は、その後のORプロセスの分割に際して極力再利用する。これにより、初期ゴールの負荷分散の高速化と通信量の低減化が図れる。

(3) 容量限界に達したプロセッサの連結

容量限界に達したプロセッサの連結は、連結された親・子のプロセッサ間で推論深さを連続させることにより実現する。連結時には、親プロセッサへの変数参照依頼が生じる。バインド情報スタックは連想アクセスされるため、外部プロセッサからの変数参照であっても、アドレス上の変換を必要とせず、通常の推論深さと変数名の2つのキーによってアクセスできる。変数の評価結果は、子プロセッサに結果として返される。

4. 連想メモリ適用の効果

並列処理環境では、アドレス系を排除した連想メモリの適用の効果が発揮される。本並列推論マシンにおける効果は、①転送に必要な実行環境の選択とガーベジコレクションの高速化、②分割ポイントの抽出、判定の高速化、③undef操作の不要化、④プロセッサ間にわたる変数参照時のプロセッサ内外のアドレス変換の不要化等である。

5. おわりに

連想メモリを用いた推論マシン：ASCAを要素プロセッサとするマルチプロセッサ型の並列推論マシンの計算機モデルとその実現手法について述べた。現在、ソフトウェアシミュレータを用いて、

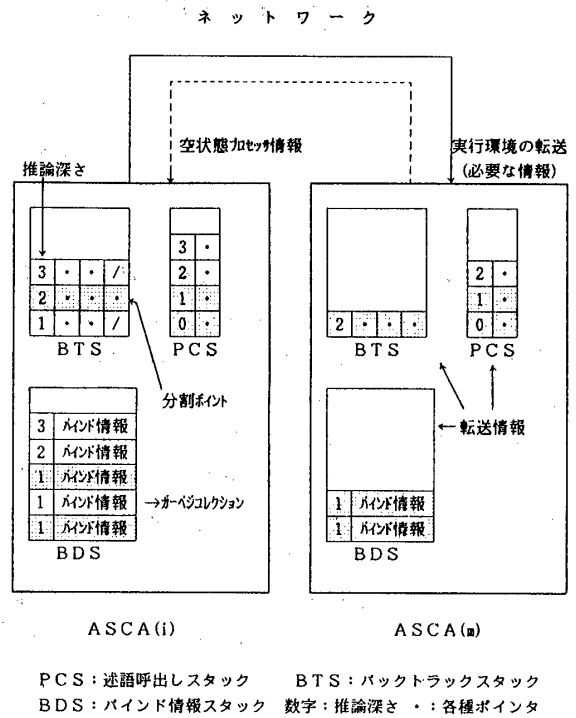


図2. ORプロセスの分割と実行環境の転送

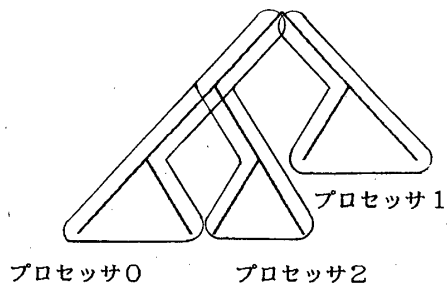


図3. 初期ゴールのORプロセスの分割

本並列推論マシンのアルゴリズムの正当性の検証を進めている。また、GHC等のAND (ストリーム) 並列型の論理型言語への適応も検討していく。

【参考文献】 (1)長沼他：連想メモリを用いた Prolog マシン (ASCA) のアーキテクチャ, The Logic Programming Conference '86, 13-1, June 1986. 連想メモリを用いた Prolog マシンとそのファームウェアインタプリタ, 第32回情処全大, 3Q-2, Mar. 1986. (2)久門他：並列推論システム - 改良型節単位処理方式, 第30回情処全大, 7C-8, Mar. 1985. (3)P. Borgwardt : Parallel Prolog using stack segments on shared-memory Multiprocessors, in Proc. 1984 Int. Symp. Logic Programming, Feb. 1984.