

1P-3

PROLOG 階層型挟み打ち並列探索法の
ALLIANT FX/4 上での性能評価

小林 和男*, 甲斐 宗徳**, 笠原 博徳*

*早稲田大学理工学部電気工学科

**成蹊大学工学部経営工学科

1. はじめに

本稿では、従来筆者等が提案している階層型挟み打ち探索法⁽¹⁾と呼ぶPROLOGの並列処理手法をマルチプロセッサ・ミニ・スーパーコンピュータALLIANT FX/4 (DSP9040⁽²⁾) 上で実現したのでその結果について報告する。この並列処理手法はOR並列⁽³⁾、⁽⁴⁾を扱うものであり、PROLOGの処理過程をAND逐次実行の条件下でOR木を用いて表現し、そのOR木を複数のプロセッサが左右から階層的に挟み打ちをする形で並列かつ独立に深さ優先探索を行うというものである。この手法ではプロセッサへの負荷割り当て単位(タスクグラニュラリティ)を大きくとることができ、負荷の割り当て制御(スケジューリング)の頻度を低減させ、スケジューリングによるオーバーヘッドを低く抑えることができる。また、プロセッサ台数以上の速度向上すなわち加速異常が得られることがシミュレータ上で確認されている。本手法は、汎用目的マルチプロセッサシステムOSCAR上でのPROLOGの並列処理手法として開発されたものであるが、それに先立って本手法の実システム上での有効性を実証するために今回FX/4上で実現したものである。

2. ALLIANT FX/4のアーキテクチャ

FXシリーズは図1に示すように、共有メモリ(物理メモリ)を中心に複数台のプロセッサを共有バス及びクロスバスイッチで結合された共有キャッシュメモリを用いて密結合した構成である。このシステム上のプロセッサは演算専用のプロセッサであるCE (Computational Element) と入出力処理を行うIP (Interactive Processor) の2種類に分けられ、各プロセッサはベクトル演算機構を装備している。FX/4ではCEを4台装備している。各CEの性能は5.0MWIPS, 11.8MFLOPSであり、モトローラ68020の命令セットに加えて、浮動小数点数に対するスカラー命令とベクトル命令を持っている。またこのシステムはコンカレンシコントロールバスと並列処理コントロールユニットを用いる並列処理命令を使って、単一のプログラムを複数のCEで処理するコンプレックスモード、一つのCEに一つのタスクを割り当てる多重ユニプロセッサモード、及び以上の二つのモードを時分割的に混在させることもできるダイナミックモードの3種の並列処理モードを実現できる。なお、本手法では多重ユニプロセッサモードを使用している。

3. 階層型挟み打ち探索法

PROLOGの処理過程は一般にAND/OR木で表される。これをAND逐次・OR並列を表現したOR木に変換する。このOR木では、根ノードからどの葉ノードへの経路(パス)の探索においても変数の共有がないため、並列かつ独立に探索可能である。

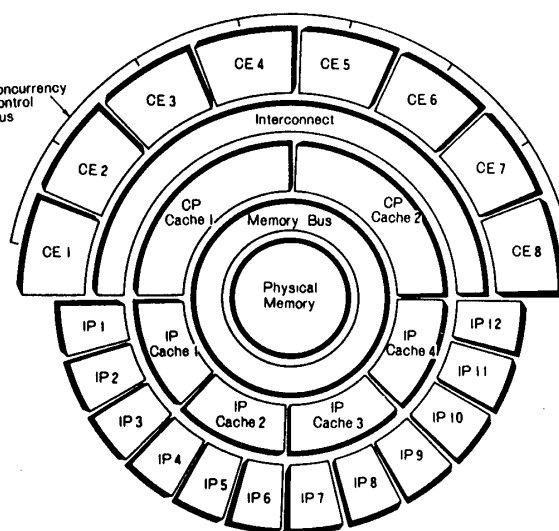


図1 FXシリーズのアーキテクチャ

本手法では、n台のプロセッサを2つのグループ、グループ1及びグループ2に分け、各グループが左右から挟み打ちの形で独自に並列探索を行う。しかし、FX/4ではプロセッサ(以下CE)を4台しか装備していないので、今回は図2のようにグループ1のみをインプリメントした。グループに属するプロセッサは、1つをリーダー、残りをスレーブとし、リーダープロセッサは通常の逐次処理と同じようにOR木の左から右へ深さ優先探索を行う。スレーブプロセッサはリーダープロセッサの探索パス上の各ノードをコントロールプロセッサから割り当てられて、そのノードを根ノードとする部分探索木を右から左へ、リーダープロセッサと挟み打ちをする形で、独立に深さ優先探索を行う。このような各プロセッサによる独立な深さ優先探索は、FX/4の場合、各CEが独立な仮想空間を持つことができるのでその上で実現される。ここでは、4台のCEのうち、CE1をリーダーCEとし、CE2~4をCE1のスレーブCEとして用いる。本手法においては、部分木の探索の割り当て等、各プロセッサの探索の管理を行うコントロールプロセッサが存在する必要があるが、この役割はリーダーCEとなるCE1に負担させることにした。また、ある部分探索木の探索が終了したかどうかをチェックするために、後述するセレクションポイント(以下SP)を導入したが、リーダープロセッサの生成するSPは他の全スレーブプロセッサが知る必要がある。OSCARでは、このSPをリーダープロセッサがブロードキャストし、他の全てのスレーブプロセッサのローカルメモリ上にSPテーブルを作ることになるが、FX/4では、CEは共有メモリを介してのみデータの授受を行うため、各C

Eの仮想空間の一部に他のCEとの共有空間を設け、そこにSPテーブルを作成した。以下、このSPを用いた並列探索過程をより具体的に説明する。SPは、

SP = (深さ、OR枝番号)

という構造になっており、ここでOR枝番号とは、現在の深さにおいてそのCEが左から何番目のOR枝を探索しているかということを示している。リーダーCEは、その深さのOR枝番号の1番の枝から番号の大きい方へと、スレーブCEは割り当てられた深さのOR枝番号の最大の枝から小さい方へと探索を進めていく。リーダーCEはOR木の探索を1つ進める毎に、SPを共有メモリ上のSPテーブルに書き込む。各スレーブCEは各自新しい探索枝の探索を開始する毎にそのSPテーブルを見て、リーダーCEの探索領域と重複していないかをチェックし、もし重複していればそのスレーブCEは探索を終了して割り当て待ち状態に入る。割り当て待ち状態にあるスレーブCEは休止するわけではなく、共有空間上のSPテーブルを参照してリーダーCEと同じ探索経路を追従し、次に割り当てられる領域の探索に必要な履歴を自己生成する。この履歴の自己再生によりデータ転送量の大幅な軽減を計ることができる。また、あるスレーブCEが割り当てられたノードにリーダーCEがバックトラックし、そのスレーブCEの部分探索木に入った場合には、スレーブCEの割り当ての深さを1つ増やし、リーダーはそのまま逐次処理を続ける。以上のように各CEは共有メモリ上のSPテーブルを使用し、各自に割り当てられた部分探索木の探索終了をチェックしながら独立な深さ優先探索を行い、OR木全体の探索が終了した時点で処理を終了する。

4. 性能評価

3.で述べたような階層型挟み打ち探索法をFX/4上で実現し、実際のPROLOGプログラム(8クイーン)を並列処理した実処理時間を図3に示す。図3から明らかなように、2台以上のCEを用いた場合には台数効果が良好に得られている。8クイーンのような全探索を行うプログラムの並列処理では、本手法を用いた場合、既にシミュレーションによる評価を行って台数効果が期待できることを報告したが、実システム上でも台数効果が得られることが確認された。また、FX/4の4台のCEのうち、1台のCEのみを用いた時、通常のPROLOGの逐次処理を行った場合には60.2秒の実処理時間が得られ、階層型挟み打ち探索を行うリーダープロセッサ(ここではコントロールプロセッサの役割も負担している)として処理を行った場合には68.5秒の実処理時間が得られた。この2つの実処理時間を比較すると、SPを生成したりコントロールプロセッサとして探索経過を管理するのに要するオーバーヘッドにより、逐次処理に比べて処理時間が14%程度増加していることがわかる。このオーバーヘッドは、SPを生成したりコントロールプロセッサとして探索経過を管理していることに起因するものである。これに対して、OSCARはコントロールプロセッサを別に設けられ、また、ブロードキャストデータ転送の機能を持っていることから、本手法の実現に当たって、逐次処理と比べて発生するオーバーヘッドをさらに低減できるものと考えられる。

5. まとめ

本稿ではPROLOGのOR並列処理手法階層型挟み打ち探索法をALLIANT FX/4上で実現し、その性能評価を行った。本手法では、タスクグラニュラリテ

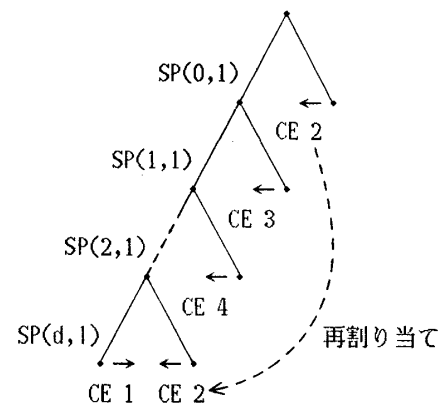


図2 OR木のプロセッサへの割り当て方式

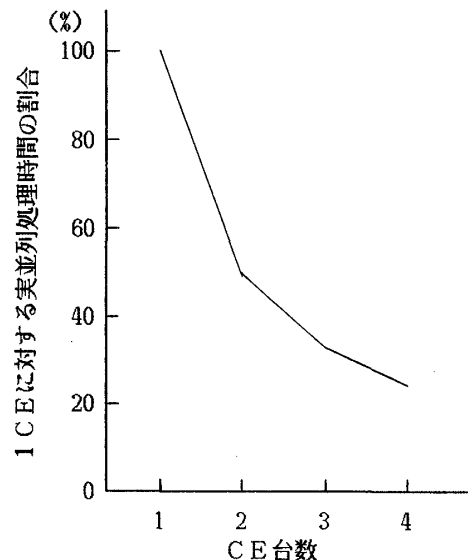


図3 FX/4上での階層型挟み打ち探索による実並列処理時間(8クイーン)

ィを大きくとれ、スケジューリングオーバーヘッド、データ転送オーバーヘッドを低く抑えることができるなどの利点があるが、実システム上でそれが有効に実現できることが確認された。現在、本手法のOSCAR上での実現を行っている。

謝辞 本研究の遂行にあたり、種々の御協力をいただいた日本アポロコンピュータ㈱に感謝致します。

参考文献

- (1) 甲斐、小林、笠原 : " 階層型挟み打ち探索によるProlog OR並列処理手法"、情報処理、vol.29、No.7、1988
- (2) Steve、榊原 : " DSP9000シリーズ (Alliant社FXシリーズ)"、Computrol、No.19、1987
- (3) 後藤、他 : " ゴール書換えモデルに基づく論理型プログラムの並列処理方式"、情報処理、Vol.25、No.3、pp.413-419、1984
- (4) 増沢、他 : " 株分け並列推論方式とその評価"、Logic Programming Conf. '86"、ICOT、pp.193-200、1986