

階層型挟み打ち法によるPROLOG OR並列処理

4T-9

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

筆者等は従来よりPROLOGのOR並列処理手法⁽¹⁾、(2)階層型挟み打ち探索法⁽³⁾を提案しその手法の有効性をマルチプロセッサ上で検討してきた。本稿では8台のプロセッサを持つマルチプロセッサ・ミニ・スーパーコンピュータALLIANT FX/80⁽⁴⁾上でその性能を評価したのでその結果について報告する。本手法は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(タスクグラニユラリティ)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。本手法は筆者等のグループが開発中である汎用目的マルチプロセッサシステムOSCARのようなアーキテクチャに適合するように設計されているが、上述のようにオーバーヘッドが低いため種々のマルチプロセッサシステム上で使用することができる。今回はそのようなシステムの例としてALLIANT FX/80上で評価を行った。

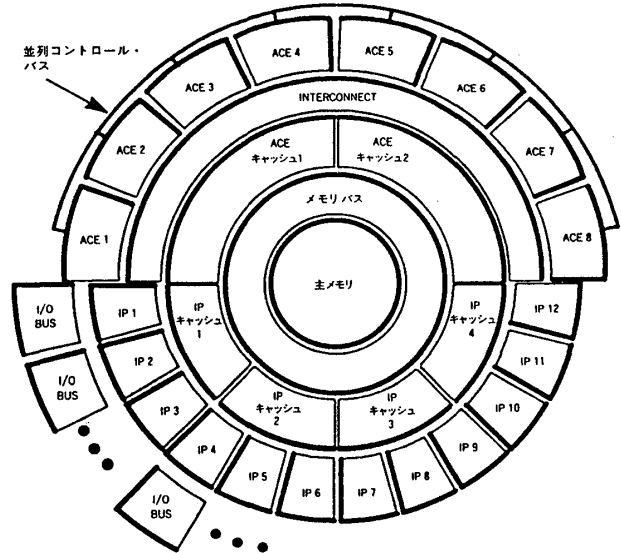


図1 FX/80のアーキテクチャ

2. ALLIANT FX/80のアーキテクチャ

FX/80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ー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。このシステムは8台の演算専用のプロセッサCE (Computational Element) と入力処理を行うIP (Interactive Processor) の2種類のプロセッサを持っており、各プロセッサはベクトル演算機構を装備している。各CEのピーク性能は23.6MFLOPS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てタスク間的高速な同期を可能にしている。

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。このAND・OR木はAND逐次・OR並列の形で考えると図2に示すようなOR木のみで表現することができる。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。

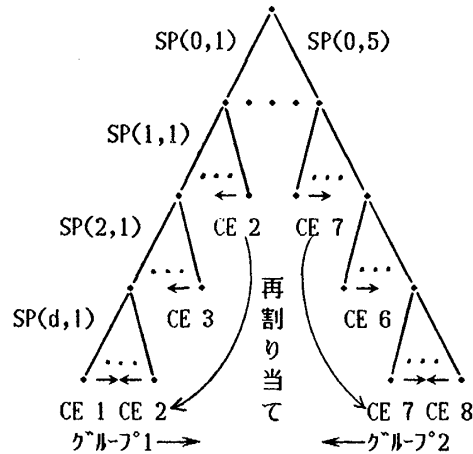


図2 OR木のプロセッサへの割り当て方式

本手法では、n台(FX/80では8台)のプロセッサ(以下CE)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。グループに属するプロセッサは、1つをリーダープロセッサ(CE1及びCE8)、残りをスレーブプロセッサとする。グループ1ではリーダーCE1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い、CE2~4の各スレーブCEはリーダーCEの探索パス上の各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CEと挟み打ちをする形で、独立に深さ優先探索を行う。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 1に負担させることにした。グループ2ではグループ1と左右全く逆に探索を行い、リーダーCE 8が右から左へ各スレーブCEが左から右へ深さ優先探索を行う。グループ2はグループ1を左右全く逆にしただけであるので以下グループ1のみの説明を行う。割り当てられ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、セレクションポイント(以下SP)を導入する。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。ここで、リーダーCEの生成するSPは他の全スレーブCE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テーブルを作成ことになるが、FX/80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、そこにSPテーブルを作成した。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。

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/80上で実現し、実際のPROLOGプログラムを並列処理した実処理時間を図3に示す。なお今回は都合によりCE台数は6台までを使用している。対象となるプログラムは8クイーン及び数式微分である。8クイーンでは全解探索を評価の指標としており、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解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。次に数式微分では最適解が一つ得られるまでの処理時間を評価の指標としている。この場合、図3から分かるように2台以上のCEを用いた場合には、CE1台の場合と比較して約1/40の処理時間となり、著しい加速効果が得られている。次に処理オーバーヘッドに関する評価であるが、8クイーンにおいて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43.1秒の処理時間であるのに対し、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)の場合には53.5秒の処理時間を要した。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、処理時間が逐次処理に比べて24%程度増加していることがわかる。これに対して、OSCARのようにコントロールプロセッサを別に持ち、さらにブロードキャストデータ転送の機能を持っているマルチプロセッサシステム上で実現すれば、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/80上で実現し、その性能評価を行った。その結果、本手法の特徴である低い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及び加速効果が確認された。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、御指導をいただいた本学成田誠之助 教授及び種々の御協力をいただいた日本アライアントコンピュータ㈱に感謝致します。

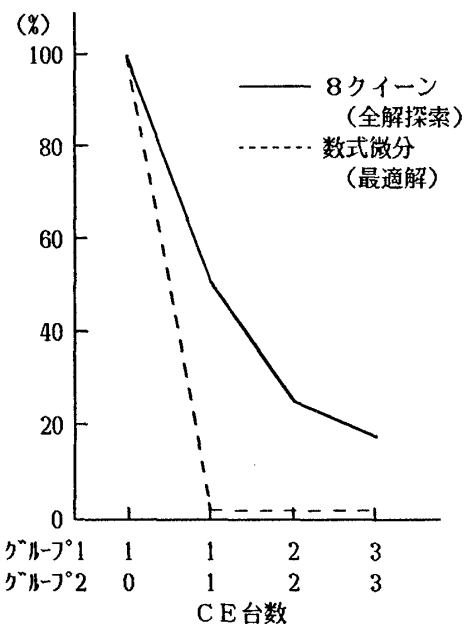


図3 FX/80上での階層型挟み打ち探索法による実並列処理時間

参考文献

- (1) 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”、情処論、Vol.25、No.3、pp.413-419、1984
- (2) 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
- (3) 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”、情処論、vol.29、No.7、pp.647-655、1988
- (4) Steve、榊原：“DSP9000シリーズ (Alliant社FXシリーズ)”、Computrol、No.19、1987