

並列推論マシン P I M — 株分け方式用コンパイラ —

久門耕一 板敷晃弘 増沢秀穂 相馬行雄
富士通株式会社

0. はじめに

我々は、通産省の第五世代計算機計画の一環として並列推論マシンに関する研究を行っている。既に我々は、株分け方式と呼ぶ逐次型処理に基づくOR並列処理方式を提案し、インタプリタによる実験的処理系（イ版と略す）を試作し、株分け処理方式の有効性を確認した(1)(2)(3)。

本稿では、処理性能を向上させる為に試作したコンパイラ版処理系（コ版と略す）の評価について報告を行う。

1. コンパイラ版株分け処理系

1.1 株分け処理法

従来のイ版では、株分けを行う際転送すべき、分割されたタスクデータの表現は、ルール番号 (RND) と呼ぶ次に実行するべき述語定義のクローズの順番を付与したサブゴールの形であった。しかし、今回のコ版では、ワレンの抽象命令セット(4) (NCODE) に基づいた実現であり、クローズインデクシングを行っている為に、実行中のサブゴールに用いている述語が何番目の定義であるかを決定する事、及び指定されたクローズから実行を始める事が困難である。又、イ版においては、内部の制御構造からサブゴールへの変換並びにサブゴールを内部の制御構造に再変換を行う為の処理時間が通信時間とほぼ同程度の時間で、株分け処理の為の時間の大きな割合を占めていた。

イ版では転送の際に一時的にUNDOを行う為に、KRENと呼ばれる変数セルのバインドの時期を示す領域を追加し、送り元が高速にUNDOを行える様にしていたが、コ版ではトレイルスタックも一緒に転送し、受け取った側でまず最初にUNDO処理を行う事により、送り元の処理量を減らした。

転送する情報としては、送り先で実行を行う為に必要な最も古い選択点までのローカル、グローバル両スタック、及び転送先の処理系が、現時点の実行環境から、再実行を行うべき環境を復元するために、トレイルスタック全節を前後処理無しで転送した。これにより、実際には使用されていない部分のローカル、グローバルスタック

のUNDOを行う事も起こるが、特に実害は無い。

1.2 処理系の構成

コンパイル実行の高速性を保証する為に、エミュレータのカーネル部は株分け実験機のCPUであるMC68Kのアセンブラにより記述した。また、入出力や株分け処理、通信部分はC言語により記述した。

ユーザが書いたプログラムのソースコードは、プログラムにより記述されたコンパイラ及びアセンブラによりNCODEを表す16進数表現に落とされる。これを株分け実験機に通信回線により転送し、コードロードにより内部にロードする。

2. 実験及び評価

従来のイ版の単体性能は、APPENDで計測した場合、約0.5 KLIPSであったが、今回のコ版では、約5 KLIPSと10倍程度の高速化がなされた。

株分けシステムのコ版の評価のために、従来からベンチマークプログラムとして用いてきたクイーン問題、並びにクローズインデクシングによりコンパイルの効果が大きいBUP構文解析プログラムを用いた。イ版1台に対するコ版の速度を表1に示す。

表1 コ処理系とイ処理系の実行速度比

問題	N-QUEEN				構文解析		
	7Q	8Q	9Q	10Q	文1	文2	文3
イ版1台	1	1	1	1	1	1	1
12台	8.0	10.4	11.6	11.9	3.3	6.9	10.5
コ版1台	4.6	5.1	5.3	5.5	20.1	21.7	21.8
12台	19.8	36.7	52.9	62.4	28.9	70.2	142

問題の大きさ

———大

———大

クローズインデクシングは逐次計算機上では、浅いバックトラックを防ぎ決定的実行部分を増すが、コ版においては、株分け後即座に失敗する無駄な株分け処理を防ぎ実行効率が向上する。

図1に2種のベンチマークによるコ版の台数効果を示す。コ版は、イ版に比較し同一の問題を解く場合には台数効果が悪くなっている(3)。

Parallel Inference Machine PIM, Compiler for KABL-WAKE Method
Kouichi KUMON, Akihiro ITASHIKI, Hideo MASUZAKA, Yukio SOHMA
FUJITSU Ltd.

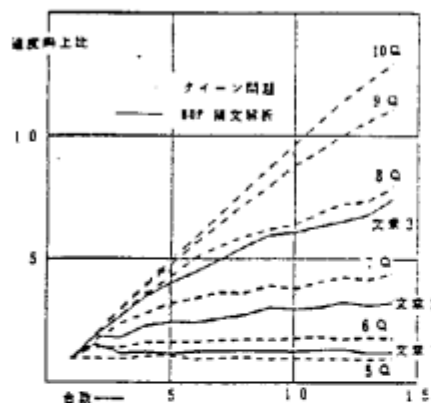


図1 コ処理系の台数効果

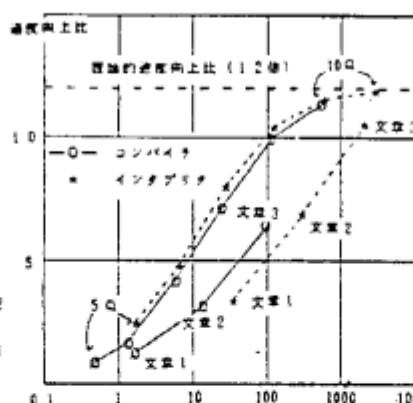


図2 問題の大きさと台数効果

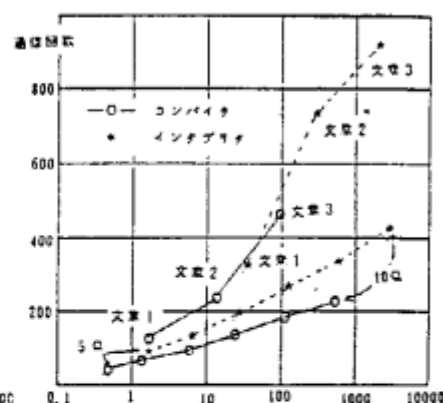


図3 問題の大きさと通信回数

図2に単位で解く場合の処理時間を問題の大きさとした場合、問題の大きさと12台での速度向上比を示す。基本的な傾向は、イ版と同様である。株分け処理マシンでは、ハードウェアの制約から、通信速度がかなり遅く、処理中の通信量の大小が実行効率を左右している。通信回数と問題の大きさの関係を図3に示す。次に、コ版によるそれぞれの問題の実行について、結果の評価を行う。クイーン問題：

図2によれば、同程度の大きさの問題ではほぼ同程度の台数効果である。クイーン問題には深い後戻りが多く、従来から台数効果の良い問題であったが、コンパイルしてもクローズインデクシングの効果が少なく、通信のオーバーヘッドがイ版と同程度であると考えられる。

しかし、図3によれば、通信回数はイ版より少なく、この点については、更に細部の測定を行う必要がある。

BIP 構文解析：

同程度の問題では、コ版の方が台数効果が良い。これは、ソースプログラム上に浅いバックトラックが多く、従来の株分け処理系では、ユニフィケーションが一回も成功しない「株」が多くあり、通信のオーバーヘッドが多かったが、コ版では、クローズインデクシングにより無駄な「株」の生成が抑えられ有効な「株」が転送される様になり効率が良くなるからと考えられる。

この様に、本質的に非決定性を多く含むクイーン問題では、コ版は従来のイ版と同程度の台数効果を持ち、決定性が多く浅い非決定性を多く含むBIPでは、コンパイルする事により効率がより良くなる。コ版においては、株分けしたタスクデータは処理せずにスタックの内容を転送する為、転送データ量がイ版に比べてクイーン問題では、約2倍に増加している。しかし、イ版では通信時間と同程度の前処理、後処理時間がかかっており、総合的な通信時間は変わっていない。

BIP 構文解析問題での文章2における各種スタックのデータ転送量の平均値は、表2に示す通り約3Kバイトである。イ版では、平均1.8Kバイトなので、1.6倍と増加している。

表2 BIP 構文解析における各種スタックの転送量

ローカル	グローバル	トレイル	計
947	1907	142	2997

文章2 12台時 単位：バイト

3. おわりに

本報告では、スタックをそのまま転送する事による、株分け処理方式を提案し、コンパイラ版の株分けシステムを作成し評価を行った。

本方式では、通信量の増加のため、通信速度の向上が重要である。

イ版の通信速度と同じ通信速度の場合、

① 処理時間がイ版とコ版で同じ程度かかる問題同士では、決定性の多いプログラムの場合には良好な台数効果が、非決定性の多いプログラムでは同程度の台数効果が得られる。

② 同じ問題では、処理時間の短縮のために通信のオーバーヘッドが相対的に大きくなり、一般にコ版はイ版より台数効果が悪くなる。

という事が分かった。

謝辞

日頃御指導及び有益な御助言を頂いている、朝樺人工知能研究部長、服部第3研究室長、並びにICOTの内田第4研究室長を始めとする関係各位に感謝致します。

参考文献

- (1)久門 他 「並列推論システムー改良型節単位方式」第30回情報処理全国大会 7C-8 PP.217-228
- (2)久門 他 「株分け処理方式の評価」第31回情報処理全国大会 2C-5 PP.43-44
- (3)増沢 他 「株分け処理方式とその評価」Logic Programming Conference '86, June 23-26th, 1986
- (4)D. H. D. Warren "An Abstract Prolog instruction Set", SRI TN309, 1983