

高並列推論エンジンPIEにおける
ゴールフレーム表現方式の評価

5B-1

丸山 勉, 酒井 保明, 田中 英彦
東京大学工学部

1. はじめに

高並列推論エンジンPIEはゴール書き換えモデルに基づき論理型言語を高並列に実行する並列推論マシンである[1]。PIEの処理単位であるゴールフレームを構成するデータ構造を新たなゴールフレームが生成されるごとに全てコピーするという従来の方法では、そのコピー等のオーバーヘッドがシステムの処理速度を低下させる危険性がある。これに対し、新たに生成された子ゴール間で親ゴールを構成するデータ構造を共有することによりこのオーバーヘッドを軽減し、より高速な処理を実現することができる[2]。この場合、共有された親ゴールに対するアクセスの集中などの問題が起こる可能性がある。本稿では、これらの点についてのシミュレーション評価について述べる。

2. ゴールフレームの表現方式

PIEにおける処理単位はゴールフレームと呼ばれ、探索木上のある時点におけるゴールを解くために必要な全ての環境(それ以降実行されるべきゴールリテラルとそれらの変数の束縛情報)から構成される。これらの環境は新たなゴールが生成されるごとにコピーされる。しかし、この環境情報の大きさは、問題によっては大きくなり過ぎ、そのコピー・転送などのオーバーヘッドがシステムの処理速度を低下させかねない。これに対し、図1に示すようにゴールフレームを定義節単位に分解し新たに生成された子ゴール間で親ゴールを共有することによりコピー・転送などのオーバーヘッドを軽減し、より高速な処理を実現することができる。この方法では、子ゴールの実行が終了すると共有されている親ゴールはコピーされ、そこに子ゴールの実行結果が書き込まれるため、共有されるデータ構造を管理するモジュールに負荷が集中する可能性がある。

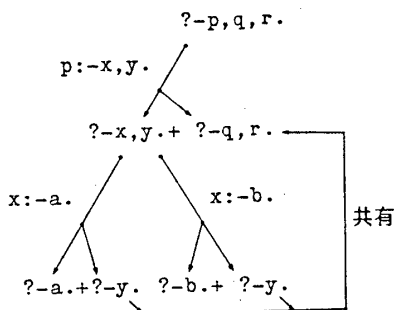


図1 ゴールの分割・共有

3. シミュレーションモデル

ゴールを定義節単位に分割・共有する方式では、ゴールの単一化において以下の操作が行われる。

- ①単一化プロセッサへのゴールの入力
- ②共有されている親ゴールへの実行結果の書き込み
- ③単一化
- ④単一化プロセッサからのゴールの出力

②は探索木の伸長を行う段階では行われない。また、従来の方式と異なり縮退の操作がないのはこの方法では不要なデータ構造の除去、変数番号の管理などが不可欠な操作ではないからである。

これらの操作において、本質的に必要な操作は単一化のみであり、他の操作は並列処理を行うために必要な操作である。従って、より高速な処理を実現するためにはこれらの処理はパイプライン処理などにより単一化時間の影に隠れることが望ましい。

シミュレーションモデルを図2に示す[2]。主要部の機能は以下の通りである。

UNI: 単一化を行う。

RED: ゴールのコピー及び子ゴールの実行結果の親ゴールへの書き込みを行う。

DMA: ゴールのコピーを行う。

UPC: UP部の管理を行う。

MMC: MM部の管理を行う。

gfm, pgfm等は単なるメモリである。

このモデルにおいて、RED, UNI, DMAは並列に動作しパイプライン処理を実現する。

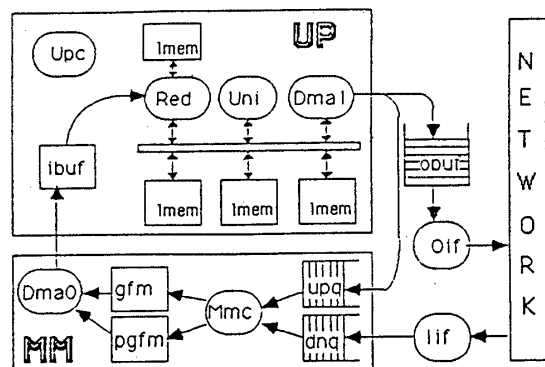


図2 シミュレーションモデル

4. シミュレーション結果

4.1 IU1台の処理速度

IU1台の場合の処理時間を表1に示す。この表から解るようにゴールを分割・共有することにより処理速度が向上している。これは、ゴール長が短くなったことと共にパイプライン処理がうまく動作していることによる。

4.2 パイプライン処理

パイプライン処理の効果を表2に示す。これは、総処理時間に対する各モジュールの稼動時間の割合である。これにより、ゴールを分割・共有することにより各ステージがほぼ均等に動作していることが解る。

4.3 負荷分散

8queensを実行した場合の経過時間に対する稼動UP数の変化を図3, 4に、実行待ちゴールの総数を図5, 6に示す。IU台数は256である。この図から解るように並列処理の環境下においてもゴールを分割・共有する方式のほうが処理速度は著しく速い。この方法では、共有されるゴールに対するアクセスの集中が起こるが、これは図6における処理の後半のピークとなって現れている。しかし、このピークはそれほど大きくはなっていない。

5. おわりに

以上ゴールを分割・共有する方式についてのシミュレーション評価について述べた。今後、より詳細な評価を行いより高速な処理を実現する方式を検討したい。

<参考文献>

- [1] Moto-oka, T., Tanaka, H., et al, The Architecture of a Parallel Inference Engine -PIE-, FGCS'84 ICOT
- [2] 丸山他, PIEにおけるゴール表現形式とそのシミュレーション評価, 第32情報全大1R-6

表1 ゴール長と処理時間

Goals		I	II
?-app(X, Y, [1, 2, 3, 4, 5]).	time	4320	3026
	length	27(30)	23(25)
?-queens([1, 2, 3, 4, 5, 6], [], X).	time	1402470	595221
	length	45(63)	19(35)
?-nrev([1, 2, ..., 9, 10], X).	time	54829	29428
	length	50(83)	26(35)

timeの単位はclock
lengthは平均と(最大値)

表2 パイプライン処理の効果

Goals	I		II	
	pipe	non-pipe	pipe	non-pipe
?-app(X, Y, [1, 2, 3, 4, 5]).	4302	4585	3062	3839
?-queens([1, 2, 3, 4, 5, 6], [], X).	1402470	2089990	595221	1020323
?-nrev([1, 2, ..., 9, 10], X).	54829	53771	29426	29678

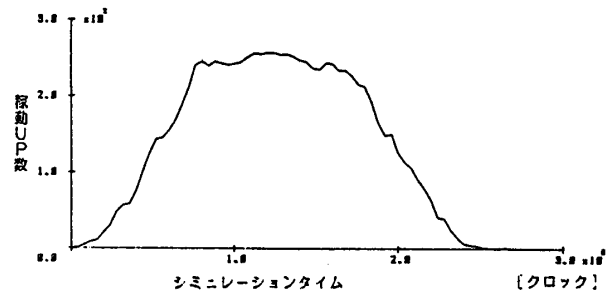


図3 従来方式における稼動UP数



図4 分割・共有方式における稼動UP数

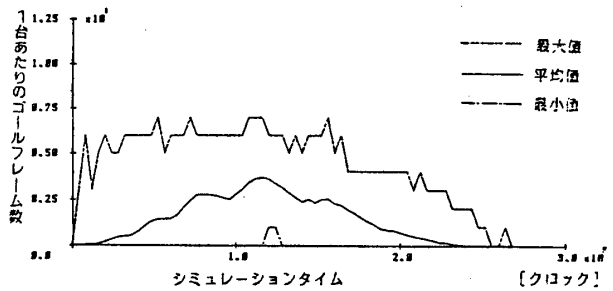


図5 従来方式におけるゴール数

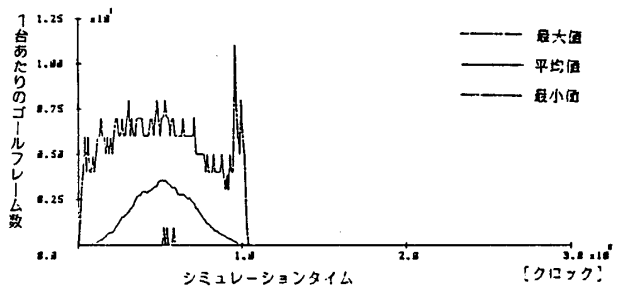


図6 分割・共有方式におけるゴール数