

PIEにおけるストリーム並列言語を
指向したネットワーク構成法山内 宗、 田中英彦
(東大 工学部)

1. はじめに

現在我々は、高並列推論エンジンPIEの開発を進めている。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推論マシンである。本稿では、GHCを始めとするストリーム並列言語を効率良く実行するためには、どのような相互結合網にしたら良いかについて検討する。

2. ストリーム並列言語

ストリーム並列言語の特徴として以下の項目が挙げられる。

- ①共有変数による値の受け渡し
- ②単一化時の同期機構
- ③ガード等によるOR並列性の制限

複数のプロセサを持つ並列推論マシンでストリーム並列言語を実行しようとする場合には、AND関係にあるゴールが複数のプロセサに分配されるため、次の点を考慮する必要がある。

- ①共有変数のアクセスと負荷分散
- ②ガード、サスペンド等のゴール間の実行制御

3. 共有変数

単一化の結果、共有変数が生じると、共有変数に相当する領域(共有変数のセル)が共有メモリ中に確保されると同時に、各ゴール中にはそこへのポインタが書き込まれ、他のプロセサに分配される。その後、単一化中に共有変数へのアクセスが生じるとプロセサはポインタをたどって共有変数のアクセスを行う。ここで大切なことは、この処理の間は単一化が中断されてしまい、共有変数へのアクセスはなるべく高速に行われなければならないということである。また、共有変数にリストなどの構造データが単一化された場合は、それらの構造データは共有メモリ中にコピーされ、共有変数のセルには、構造データへのポインタが書き込まれることになる。従って、プログラム実行中にできる構造データは全て共

有メモリ上にとられることになる。また、ストリーム並列言語では共有変数の値によってゴールのサスペンド、アクティブイト等の制御が行われるので、それらの実行制御を行う機能も共有メモリに要求される。

4. 負荷分散

並列計算機では、各プロセサにどのように負荷を分配するかが非常に重要である。動的に負荷分散する戦略をとるならば、各プロセサの負荷状況を見てどのプロセサに負荷を分配したら良いかを決めるという方式が基本であるが、その場合、各プロセサの負荷をどのように評価するかが重要である。

高並列推論マシンPIEにおいては、各プロセサの負荷量はユニット中のゴールの数で決めている。但し、ストリーム並列言語においてはサスペンド状態であるゴールの数が非常に多く、これらのゴールの数を考慮に入れずにゴールの分配(すなわち負荷の分配)を行うと、そのサスペンド状態のゴールが実行可能状態になったときにプロセサの負荷が大きくなっていく危険がある。そこで、実行可能なゴール数が同じ場合には、サスペンド状態のゴール数の少ないプロセサにゴールの分配を行う必要がある。

5. PIEにおける2種のネットワーク

前述のように、ストリーム並列言語を指向したPIEにおいては、各プロセサ間の通信として大きく分けて「共有メモリへのアクセス」と「ゴールの分配」の2種が考えられる。そして、「共有メモリへのアクセス」は、各プロセサが単一化を中断して行うので非常に高速な応答と高スループットを必要とする。一方、「ゴールの分配」は高スループットの必要はあるが、それ程速い応答は必要ない。このように、この2種類の通信は特性が大きく異なっているので、パフォーマンス、コストの両面から考えて、別々のネットワークとしたほうが良いと考えられる。

そこで、PIEのネットワークは、共有メモリ網(SMN)とゴール分配網(DN)の2つを用意している。

①ゴール分配網(DN)

前述のように、ゴール分配(すなわち負荷分散)は各プロセサの負荷状況を評価して、負荷の軽いプロセサへゴールを送り付けるという方式をとる。PIEのゴール分配網(DN)は、オメガ網であり、通信経路とは逆向きに各プロセサの負荷情報を流し、各スイッチング・ユニット(SU)が、それを比較して最も負荷の軽いプロセサへの経路を設定するという「自動負荷分散網」である。そして、通信経路を逆向きに流れる負荷情報としては、各プロセサのゴール数を用いる。

②共有メモリ網(SMN)

前述のように、共有メモリへのアクセスは、単一化を中断して行われるので、高速な応答を必要とし、また、各プロセサからのアクセスの集中も考えられる。そこで、PIEでは、1台の共有メモリと1台のプロセサを1つのユニットとして、それらを多数台接続する方式をとっている。このようにすることによって、プロセサは対となっている共有メモリへはSMNを介さずに高速にアクセスすることができるため、共有変数の配置をうまく行えばSMNのトラフィックをかなり減らすことができる。共有メモリとプロセサをペアとしたユニット間を結ぶネットワークは、高速な応答が必要であるが、現時点ではDNと同様の多段結合網かクロスバーを考えており、詳細なシミュレーションをする予定である。

6. キャッシュ・メモリ

共有メモリへのアクセスは高速性が要求されるので、キャッシュ・メモリを採用することが考えられる。以下に、キャッシュ・メモリの目的を示す。

①共有メモリ・アクセスの高速化

②SMNのアクセス競合を軽減

しかし、マルチ・キャッシュのコヒーレンスを保つための制御が複雑になり、オーバーヘッドが大きくなる可能性がある。また、シミュレーションより、キャッシュ・メモリのヒット率がかなり低いことがわかっているので、より詳細なシミュレーションでキャッシュ・メモリの有効性を検討する必要がある。

7. 階層構成

大規模の並列計算機を構成する場合には、どのように各プロセサを接続するかが重要である。数百、数千という規模の並列計算機をクロスバーや完全網で接続するのは非常に無理があるので、次の2つのアプローチが考えられる。

①メッシュ系の相互結合網

②相互結合網の階層化

メッシュ系の相互結合網は、プロセサ間の距離が離れるにつれて緩やかに通信コストが増えていくので、見方によっては非常に細かい(アナログ的な)階層を持った相互結合網とみなすことができる。階層構成にした場合は、クラスタ間の通信は遅延が大きく、通信路のバンド幅は狭いので、なるべくプログラムのローカリティを生かしてマッピングすることによってクラスタ間の通信を減らしたい。その場合、メッシュと階層構成とはどちらがプログラムの特性を生かしたマッピングが容易であるかを詳細に検討する必要がある。また、共有メモリへのアクセス・トラフィックを減らすことと負荷をうまくプロセサに分配することは裏腹の関係になることが多いので、ある程度のトレード・オフは避けられない。

8. 今後の予定

多くの例題プログラムがどのように動くかを検討して、各種の問題の妥協点を見出す必要がある。特に、負荷分散と共有変数の配置等の処理方式とネットワークの構成は密接に関連しているので、より詳細なシミュレーションをして構成を決定していく必要がある。

<参考文献>

- [1] 坂井、小池、田中、元岡：“動的負荷分散を行う相互結合網の構成”，情報学論, 27, 5, pp. 518-524 (昭61-5)
- [2] 宮崎、瀧：“Multi-Psi におけるFlat GHCの実行方式”，Proc. of Logic Programming Conference '86, ICOT, 1986.
- [3] 垂井、田中：“並列推論マシンにおけるストリーム並列言語の実行方式の評価”，Proc. of Logic Programming Conference '87, ICOT, 1987.
- [4] 丸山勉：“論理型言語の並列処理方式の研究” 東京大学学位請求論文1986年12月