

並列計算機の結合形状評価用シミュレータ

1T-6

武本 充治

松本 尚

平本 敬

東京大学理学部情報科学科

1 はじめに

高速計算機に対する要求が年々高まりつつあるのは事実である。しかし、既存の逐次計算機による高速化だけではもはやその要求に答え切れなくなっているため、当然の帰結として並列計算機に解を求めることとなる。現在のアーキテクチャ的な研究対象は単なる並列計算機ではなく、プロセッサ要素が数千台以上のいわゆる超並列計算機となっている。内部相互結合網は高い並列度で効率の良い実行を行う上では重要な要素である。そこで、この分野の研究も盛んに行われており、各種形状 [1][2][3] が提案されている。結合形状やフロー制御方式など逐次計算機の場合には存在しなかった要素についての研究も行われている [4]。

並列計算機では通信遅延が全体の性能に影響を及ぼす場合がある。最近の並列計算機では通信のためのプロセッサを専用に設けることで、通信と演算のオーバーラップを行い、これにより通信遅延の隠蔽の実現している。また、アプリケーションに関しても通信と演算をオーバーラップするようにコードを書き換えれば通信遅延の隠蔽に効果がある。

以上の状況を踏まえ、相互結合網に関する要素も評価できるシミュレータ [5] を作成し、通信遅延隠蔽の意味での最適化を施したアプリケーションを用い、結合形状の変化の影響を調べた。

2 シミュレーション方式

集中共有メモリでは実際に構成するシステムの規模に限界があるので、シミュレートする並列計算機のシステムのメモリ構成は分散メモリとした。現時点では分散共有メモリの機構は実現していない。各ノードは CPU の他にインテリジェントな結合網通信コントローラ (NIP: Network Interface Processor)、同期ビットを付加したメモリ [6]、及び、スヌープキャッシュから構成される (図 1)。結合形状は [4] のように構文を持ったノード接続記述言語を用いるのではなく、各ノードの接続を明示的にすべて記述した結合形状ファイルで指定するものとした。また、今回のノード間の同期はメモリシステムに付属し

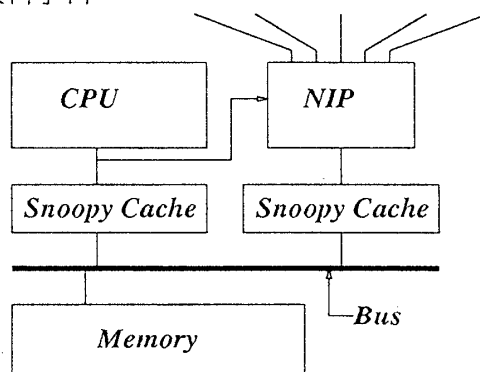


図 1: ノード構成図

た同期ビット [6] を使用し、データの通信と不可分に行われる。

CPU が NIP に発行する通信命令の引数はメモリのポインタとサイズである。CPU が通信命令を発行すると、通信に関する動作はすべて NIP が行うものとした。送信命令ならば NIP がメモリからデータを読み、パケットを生成し、他のノードへ転送する。受信命令を受け取った後、目的パケットが到着していればメモリに同期ビットつきでデータを書き出す。CPU がデータを必要とした時にデータが未到着ならばメモリ側でブロックされるが、他の通信に関するイベントで CPU がブロックされることはない。他のノード宛のパケットはルーティングアルゴリズムにしたがって転送する。

使用した CPU の命令は R4000 に準拠している。通信命令はライブラリで提供され、アプリケーションは C で記述される。市販の最適化コンパイラを用いてコンパイルを行い、当研究室で開発した変換システムを使用すれば、シミュレータ上で実行可能となる。

3 通信遅延隠蔽

通信に関する機能を独立に実行する NIP の存在を仮定すれば、演算と通信のオーバーラップは可能である。NIP を独立に動作させることでノード間通信遅延の隠蔽効果が得られる。また、プログラムコード自身も他のノードに割りつけてあるデータをプリフェッチするような書き換え、つまり、通信遅延隠蔽の意味の最適化が施されるべきである。

このアプリケーションの最適化は通信・同期命令の位置を変更する程度のもので効果を得られることをシミュレータを使用した実験により示す。数値計算をはじめとする多くのアプリケーションではノード間に跨るデータ依存が静的に分かる場合が多いため、この最適化は有用であるといえる。

Network Topology Simulator for Massively Parallel Computers
Michiharu TAKEMOTO Takashi MATSUMOTO Kei HIRAKI
Department of Information Science, Faculty of Science, the
University of Tokyo

7-3-1 Hongo, Bunkyo-ku, Tokyo, 113, Japan

E-mail: {skelton, tm, hiraki}@is.s.u-tokyo.ac.jp

4 実験と結果

今回行った実験について述べる。使用した結合網は3種類でいずれも64ノードで、リング(図2中:r)、ハイバキューブ(図2中:h)、2次元メッシュ(図2中:m)である。パケットの転送に関してはストアアンドフォワードでバチャルカットスルーは行わないものとした。ノード間の転送能力に関しては1ワード(4バイト)を転送するのにかかる時間をCPUのクロックで4~32クロックと変化させた。

シミュレータ上で実行させるアプリケーションとしてRed-Black SORを2種類C言語で記述した。通信と同期に関する最適化を行わずに他のノードに存在しているデータが必要となった時に通信命令を発行するもの(図2中:L1)と通信と同期に関してCのソースレベルでの最適化を行なったもの(図2中:L2)である。最適化は2つの点に注目して行った。1つはループ内に存在する通信命令をできるだけ前に移動するというもので、もう1つは同期ビットを使用してデータの待ち合わせをする位置をできるだけ後ろに移動するというものである。両コード共に2次元空間を格子点状に分割し、プロセッサ1つ当たり16×16格子点をブロックマッピングした。通信パケットのデータ部の大きさは2ワード(Cの倍精度小数)とした。

図2に実験結果を示す。横軸はノード間で1ワードのパケットを送るの時に必要な時間(CPUクロック)で縦軸が総実行時間(CPUクロック)である。

通信・同期に関して最適化を行っていない場合(図2中:L1)ではリング状結合の場合だけ大きく差が生じる。SORというアプリケーションでは1つの格子点の更新には上下左右の4方向の更新点のデータを必要とする。ブロックマッピングのため、1つのプロセッサに割り当てられた格子点の境界部分の更新を行う時のみに通信を必要とする。あるプロセッサが通信する可能性のあるプロセッサは4つである。このため更新に最適な形状は2次元メッシュである。リングは通信が2方向のノードとしか行えないため、パケットのノード間転送に時間がかかる場合は結合形状の影響が生じる。

最適化を施すと上記のどの3形状においても約2倍の性能向上が見られる(図2中:L2)。また、転送能力の変化の影響も殆んどなくなる。そして、通信に関係のない演算を実行している間に通信を行えるため結合形状の差が殆んどないまでに吸収されてしまっている。

5 おわりに

並列計算機に重要となる内部相互結合網の結合形状をアプリケーションの通信・同期の最適化と合わせて評価した。その結果Red-Black SORでは最適化を行えば結合形状の差を吸収できることを示した。また今後は個々のアプリケーション毎の通信・同期の各種最適化を行い、各種結合形状の上で評価を行っていく予定である。

参考文献

- [1] 楊 愚魯, 天野 英晴, “超並列向きプロセッサ結合網の提案,” 電子情報通信学会技術研究報告, CPSY92-52, vol. 92, no. 290, pp. 51-57, Oct. 1992.
- [2] 岩崎 一彦, イゼリ クリスチャン, 佐藤 裕二, “超並列計算機用VLSIに適した結合網の一提案,” 電子情報通信学会論文誌, vol. J75-D-I, no. 8, pp. 583-591, Aug. 1992.
- [3] 奥川 峻史, “超並列向き de Bruijn (DB) 網の諸特性,” 電子情報通信学会論文誌, vol. J75-D-I, no. 8, pp. 592-599, Aug. 1992.
- [4] 柴村 英智, 久我 守弘, 末吉 敏則, “超並列計算機のための相互結合網シミュレータ,” 並列処理シンポジウム JSPSP '93 Joint Symposium on Parallel Processing 1993 論文集, pp. 159-166, May 1993.
- [5] 武本 充治, 松本 尚, 平木 敬, “レイテンシ隠蔽における結合形状の評価,” 電子情報通信学会技術研究報告, SWoPP 93, Aug. 1993.
- [6] Matsumoto, T., T. Tanaka, T. Moriyama, and S. Uzuvara, “MISC: a Mechanism for Integrated Synchronization and Communication using Snoop Caches,” in *Proceedings of the International Conference on Parallel Processing*, vol. I, pp. 161-170, 1991.

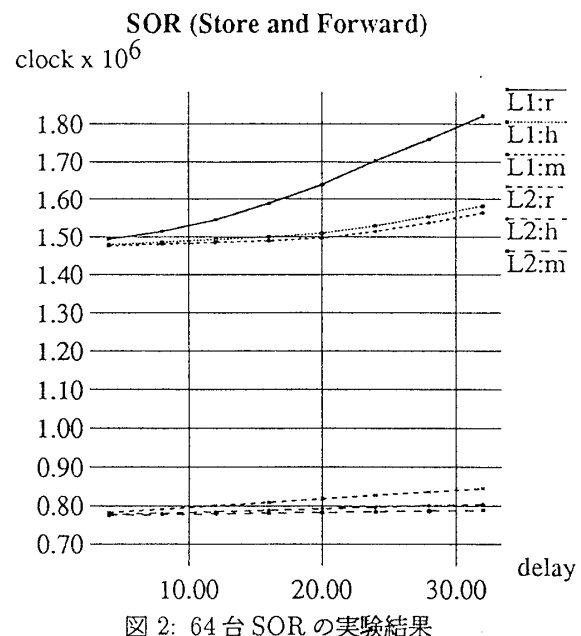


図2: 64台 SOR の実験結果