1P−3 多重命令流プロセッサを用いる多段ネットワーク結合共有メモリ型並列機のシミュレーションによる性能評価

数藤義明　田胡和哉*　永松礼夫　出口光一郎　森下巖
東京大学工学部

1 はじめに

多段結合ネットワークを用いた共有メモリ型マルチプロセッサ・システムでは，メモリアクセスがネットワークを通して行われるために，大きなアクセス遅延が生ずる．この遅延をキャッシュメモリを用いることによって解決することが試みられてきたが，キャッシュコヒーレンス問題や，キャッシュミスによるアクセス遅延の問題は依然として残っている．しかし，このアクセス遅延の問題は，システム全体を一つのパイプライン処理系とみなし，単一のプロセッサ中で複数の命令流を実行させる多重命令流プロセッサを用いることにより解決できる．本稿では，このような多重命令流マルチプロセッサ・システムをシミュレーションによって性能評価し，この多重命令流プロセッサが多段結合ネットワークを用いた共有メモリ型マルチプロセッサ・システムに有効であることを示す．

2 多重命令流プロセッサ[1)]

単一プロセッサ中に複数の命令流をパイプライン実行するプロセッサを多重命令流プロセッサと呼ぶ．各命令流の個々のインストラクションは数段のパイプラインで実行されるが，このときメモリフェッチを送信段と受信段とに分割し，その間に数クロックの余裕が出来るように実行する(図1)．この数クロックの間，プロセッサは他の命令流の実行を行なっているのでプロセッサ内部の資源の無駄にはならない．またこの間にメモリアクセスがネットワークを通して完了することが可能である．このために多重命令流プロセッサを用いれば，多段結合ネットワークのアクセス遅延の問題が解決できる．

3 シミュレーションの概要

実現したシミュレータでは，CPUにSPARCチップのインストラクションを5段のパイプラインで実行可能な多重命令流プロセッサを用いた．キャッシュメモリには，4ワード先読み可能なコードキャッシュを用い，相互結合網にはオメガ結合型のネットワークを用いた．比較実験用にネットワークを通さずに共有メモリにアクセスできるモード(パラコンピュータモード)も設けた．

このシミュレータ上で実行させるアプリケーション・プログラムを作成するにあたり，以下の様な並列実行用の関数を用意した．

V_start() 空きプロセッサ(命令流)の割当て要求命令

V_wait() 割当てられた全てのプロセッサ(命令流)の実行終了待ち命令

f_a() Fetch and Add 命令

これらの関数はC言語のライブラリとして提供され，それをリンクすることにより現在使用可能なSPARC用コンパイラによって並列アプリケーション・プログラムを作成した．

4 性能評価

並列化されたライフゲームのプログラムを実際にシミュレータ上で動作させた結果を以下に示す．実験したライフゲームのフィールドの大きさは64*64，またネットワークスイッチのバッファ数は1入力に付き8個，ネットワーク全体は2入力2出力のスイッチで構成した．キャッシュメモリには十分なサイズである2KBを用意した．

4.1 仮想的なスケジューリング

まずプロセッサ(命令流)のスケジューリングを行なうスケジューラの実現方法による影響を排除する為に，仮想的なスケジューラをシミュレータ上に用意して実験を行なった．この結果，スケジューリングに要する時間が無限小となり，実行すべき仕事を各命令流に効率よく割当てることが出来るようになる．この場合のプロセッサ数に対する総実行クロック数を図2に，一つのプロセッサ内の命令流数に対する総実行クロック数を図3にしめす．

このような仮想的スケジューリングを行った場合には，多重命令流プロセッサが期待された性能をしめし，理想的なパラコンピュータにほぼ近づけることが分かる．さらに一つのプロセッサ内の命令流の数は，ある最適値を持つことが分かった．これは，命令流の数を多くすることによって，逐次実行される部分では実行時間が延びるが，ネットワークを通してメモリアクセスするアクセス遅延の影響が少なくなる．この二つが原因で，実行時間が最少となる命令流数の最適値が存在することになる．

4.2 ソフトウェアによるスケジューリング

スケジューラの実現方法の一つとして，プロセッサ(命令流)のHALT命令と，あるプロセッサから異なるプロセッ

Performance Evaluation through simulation of a Multistage-Network Shared-Memory Parallel Machine with Multi-Instruction-Stream Processors
Yoshiaki Sudo,Kazuya Tago*,Leo Nagamatsu,Koichiro Deguchi,Iwao Morishita
University of Tokyo
*現在所属，日本IBM東京基礎研究所

サへとネットワークに通して運ばれる反射パケット割込みによるプロセッサ(命令流)の起動という最も簡単なスケジューリング機能だけをシミュレータ上に用意し，スケジューラをソフトウェアで作成した．この場合のプロセッサ数に対する総実行クロック数を図4にしめす．

このように，非常に簡単な機構とソフトウェアによるスケジューリングを行なった場合でも，前節の結果と同じ様に，多重命令流プロセッサは単一命令流プロセッサと比較して，十分な速度を得られることが分かった．

5 まとめ

多重命令流プロセッサを用いたマルチプロセッサ・システムの構成を設計し，実際のアプリケーション・プログラムを動作させるシミュレーションを行なった．仮想的スケジューリングを行った場合やソフトウェアによってスケジューラを実現した場合でも，このシステムが理想的なパラコンピュータにほぼ近づけることが分かった．この結果から，実現の容易なコードキャッシュだけを用いたとしても，多重命令流プロセッサは，単一命令流プロセッサと比較して，多段結合ネットワークを用いた共有メモリ型マルチプロセッサ・システムにはかなり有効であることが確認された．しかし，多重命令流プロセッサでは，単一命令流プロセッサと比べて個々の命令流の実行時間は数倍になる．その点を考慮してシステムの設計を行わなければならないであろう．

今後の課題としては，他のアプリケーションに対しても同様のシミュレーションをおこない，多重命令流プロセッサの有効性を示すことや，今までとは異なった並列実行プリミティブを用意して，プロセッサの起動方式の違いなどによる影響を調べる必要がある．

参考文献

1) 森下巖: 多段結合ネットワークを用いる超並列マシンのためのパイプライン化 MIMD プロセッサ, 情報処理学会誌, Vol.31, No.4, pp,523-531(1990).

2) Sun Microsystems,Inc.:The SPARC^{TM} Architecture Manual,Sun Microsystems,Part No:800-1399-07(1987).

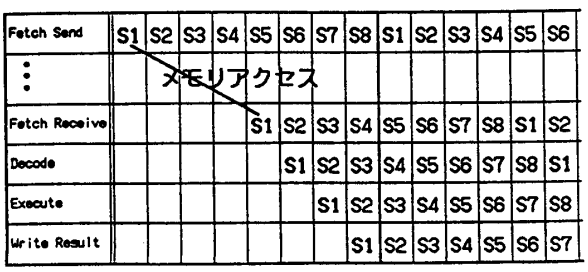

図1 多重命令流プロセッサのパイプライン実行(命令流数8個)

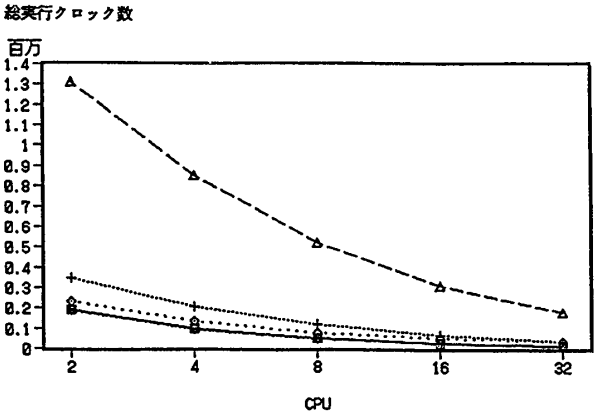

図2 仮想的スケジューリング
プロセッサ数に対するトータルサイクル数

□ 多重命令流キャッシュ付き　＋単一命令流キャッシュ付き
◇ 多重命令流キャッシュ無し　△ 単一命令流キャッシュ無し
× 多重命令流パラコンピュータ　▽ 単一命令流パラコンピュータ

CPU=2 命令流 =14　CPU=4 命令流 =16　CPU=8 命令流 =18
CPU=16 命令流 =20　CPU=32 命令流 =22

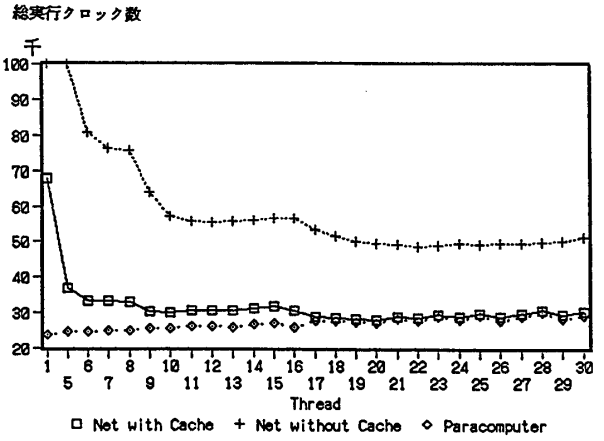

図3 仮想的スケジューリング
命令流数に対するトータルサイクル数(CPU=16)

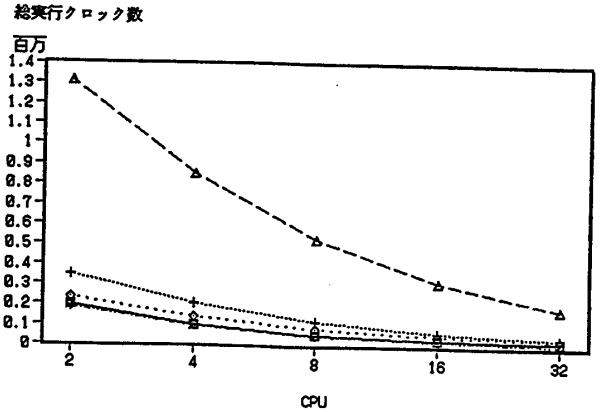

図4 ソフトウェアによるスケジューリング
(凡例は図2と同じ)