

マルチプロセッサ・ワークステーションでの  
並列FFTのキャッシュ性能

5X-10

Cache Performance of Parallel FFT on a Multiprocessor Workstation

堀口 進 中田 武男

東北大学工学部情報工学科

日本IBM東京基礎研究所

1. はじめに

FFTは、光導波路の設計、音声認識、画像処理などの様々な分野で用いられている。計算機の応用分野が広がるとともに、処理されるデータの数は、益々膨大になってきた。その結果、超高速フーリエ変換を目指した並列FFTアルゴリズムが盛んに研究されてきた。我々は、10台のIntel 80386からなるマルチプロセッサ・ワークステーションTOP-1[1]に並列FFTアルゴリズムをインプリメントし、性能評価を行った[2]。本報告では、キャッシュ・プロトコルによる並列FFTの処理性能について比較検討した結果について述べる。

2. システム構成

マルチプロセッサ・ワークステーションTOP-1(Tokyo Research Parallel Processor-1)は、10組のIntel 80386とFPU(Weitek 1167, Intel 80387)より構成される。それぞれのプロセッサは、128kBのスヌープ・キャッシュメモリを有し、ソフトウェア開発・移植の容易性を考えた共有メモリ型の密結合の単一バスアーキテクチャを採用している。バスは、64ビットの2way-interleave方式で、128kBのキャッシュメモリを各プロセッサが有している。マルチプロセッサ・ワークステーションのハードウェア構成を図1に示す。共有メモリは、32MBの容量であり、187.5nsのサイクル・タイムで使用される。外部記憶として300MBの固定ハードディスク4基が実装されており、4MBのディスク・キャッシュメモリを介して共有メモリに結合されている。

並列・分散処理のオペレーティングシステムとして、カーネギーメロン大学のMachが有名である。マルチプロセッサ・ワークステーションTOP-1の並列オペレーティングシステムは、従来のプロセスに比べスイッチングの軽いスレッドモデル[3]とUNIX<sup>TM</sup> System Vの共有変数を用いたプロセスモデルがある。今回、プロセスモデルを用いて並列FFTアルゴリズムをインプリメントとした。

3. キャッシュプロトコル

TOP-1のキャッシュ・システムには、更新型(Update)と無効型(Invalid)のスヌープ・プロトコルの2方式が同時にサポートされている。更新型プロトコルは、共有データへのライト・オペレーションが生じた時、システム中の他の全てのコピーを更新することによってキャッシュ間のデータ一貫性を保つ。一方、無効型プロトコルは、共有データに対するライト・オペレーションが生じた時に、システム中の全てのコピーを無効化することによって、キャッシュ間のデータ一貫性を保つ。両者のプロトコルの状態遷移図を図2に示す。

Cache Performance of Parallel FFT on a Multiprocessor Workstation

Susumu HORIGUCHI:Dept. Information Eng., Tohoku University, Sendai 980, JAPAN

Takeo NAKADA :Tokyo Research Laboratory IBM Research Chiyodaku3-5-19, Tokyo 120 JAPAN

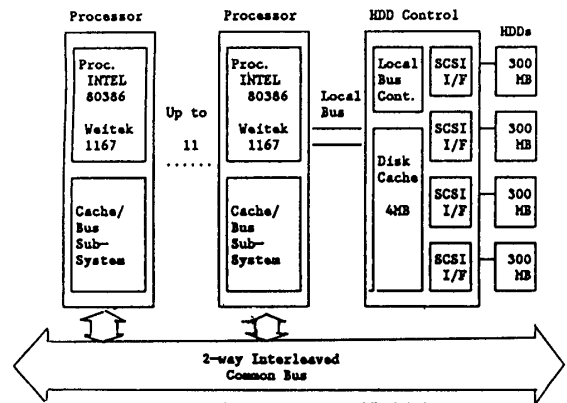


図1 システム構成図

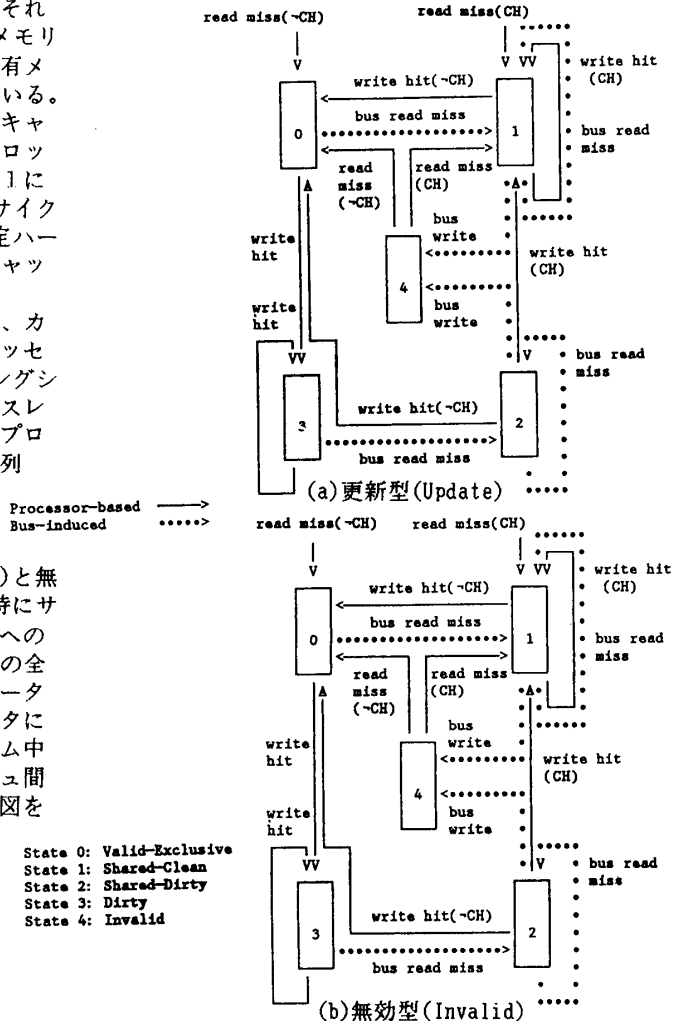


図2 更新型と無効型スヌープ・プロトコル状態遷移図

<sup>TM</sup> UNIXは、米国AT&T社の登録商標である。

4. 並列FFTの性能評価

2次元FFTは、行方向と列方向をP個の独立のNポイントFFTを行うプロセスにより並列計算される。行方向と列方向、それぞれのFFTの間で共有変数を用いてプロセス同期を実現している。

図3に両者のプロトコルで2次元データ(NxN)を並列FFT実行した場合の処理速度比を示す。ここで、処理速度比は1個のプロセッサで実行した処理時間を、P個のプロセッサで並列計算した処理時間で割った値である。データ数が大きい場合は、Pが8まで、ほぼプロセス数に比例した処理速度比が得られることが分かる。実際、TOP-1のプロセッサの内、1台は並列オペレーティングシステム専用で、8台が複数のプロセスを並列実行する。並列FFTの処理速度比は、データ数Nが大きい場合、P=4、8で無効型プロトコルの方が更新型プロトコルより高い。P=2では、両者の違いは見られない。

図4に、両プロトコルのキャッシュ・ヒット率を示した。データ数は512x512、プロセス数Pは8の場合で、インストラクション、データ・ライトおよびデータ・リードの各々のプロセス終了までの平均キャッシュ・ヒット率を示した。プロセッサ#0は、システム専用であり、プロセッサ#1から#8までがプロセスを並列実行している。両プロトコルともインストラクションは、ほぼ100%に近く、データ・リード、ライトとも高いキャッシュ・ヒット率が得られていることが分かる。

5. 結論

本報告では、マルチプロセッサ・ワークステーション上に並列FFTをインプリメントし、キャッシュ・プロトコルによる処理性能について検討した。その結果、両プロトコルとも高いキャッシュヒット率が得られる事が分かった。また、データ数Nが大きい場合、無効型プロトコルの方が更新型プロトコルより高い処理速度比が得られることがわかった。更に、マルチプロセスのスケジューラによるオーバーヘッドも比較的少なく実際のマルチプロセッサシステムで並列処理できることが分かった。

現在、並列ソーティングの処理速度、キャッシュ・ヒット率に関して性能評価を行っている。

謝辞 本研究のため便宜をはかって下さいました、東北大学木村正行教授、日本IBM東京基礎研究所鈴木則久所長ならびにサポートして下さったワークステーション・グループの皆さんに感謝致します。

参考文献 [1] 清水他「高性能マルチプロセッサ・ワークステーションTOP-1」並列処理シンポジウムJSP'89, pp.151-162(1989)  
 [2] 堀口、中田「マルチプロセッサワークステーションでの並列FFT」並列処理シンポジウムIEICE (1989).  
 [3] 穂積「TOP-1オペレーティングシステム」情報処理研88-OS-40 (1988).

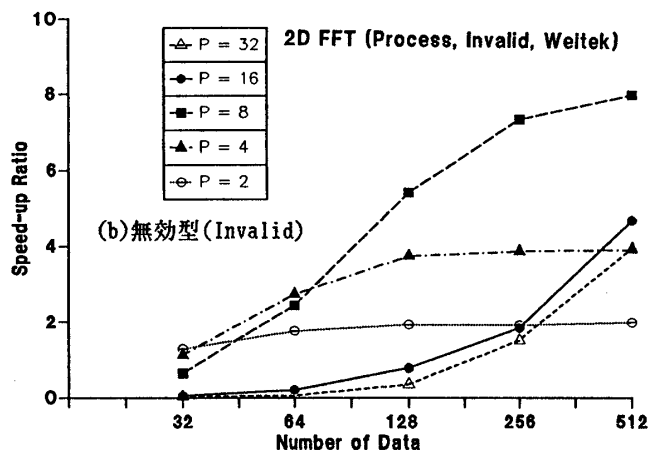
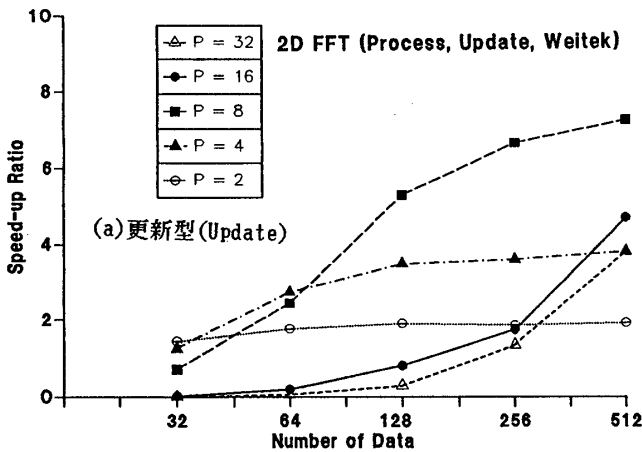
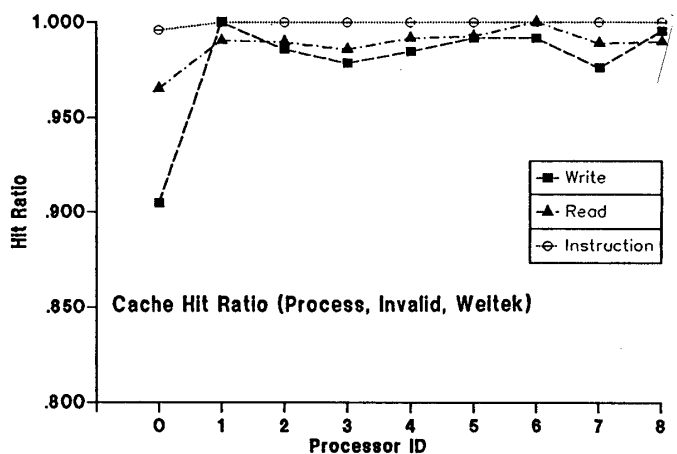
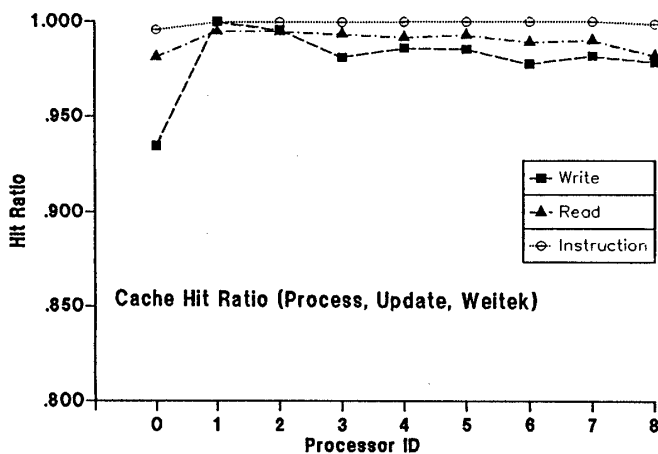


図3 二次元FFTの並列計算処理速度向上比



(a)更新型(Update)

(b)無効型(Invalid)

図4 無効型と更新型プロトコルのキャッシュ・ヒット率