

## プロセッサ間交信評価システム MILL の並列計算機への実装\*

2B-3

遠藤 浩<sup>†</sup> 寺澤 卓也<sup>†</sup> 鬼頭 宏幸<sup>‡</sup> 服部 大<sup>‡</sup> 山本 淳二<sup>‡</sup> 天野 英晴<sup>‡</sup>(株)日立製作所 公共情報事業部<sup>†</sup>東京工科大学 情報通信工学科<sup>‡</sup>慶應義塾大学 理工学部<sup>‡</sup>

## 1 はじめに

並列計算機シミュレータは並列計算機アーキテクチャや並列アプリケーションの詳細で且つ実的な性能を検証する手段として用いられる。しかしながら、対象とする並列計算機システムのプロセッサ規模に比例してシミュレーションに要する時間が増大するため、高速化に対する要求は強い。並列処理による高速化はそのうち最も有望な方法の一つである。

プロセッサ間交信評価システム MILL(Multiprocessor system Instruction Level simuLator) は数十台規模のプロセッサ装置を持つバス結合型並列計算機を CPU のインストラクションレベルでシミュレートするシステムであり、キャッシュシステムやプロセッサ間交信機構の評価に用いられている [1]。しかしながら、インストラクション毎にプロセッサをエミュレートするためにシミュレーションに莫大な時間が費されてきた。そこで、本研究では処理時間の短縮を目的とした並列化を行ない並列計算機に実装を行なう。また、効率の良い並列処理を実現するための鍵となる時刻管理機構には、分散時刻管理機構であるコンサーバティブ法の間合せに基づく方法を改良したアルゴリズムを採用し、時刻管理のオーバヘッド低減を試みた。

本報告では MILL の並列計算機への実装と評価について述べる。

## 2 並列計算機シミュレータ MILL

MILL は共有メモリを持った単一バスのバス結合型並列計算機を対象とした並列計算機シミュレータである。シミュレートする各 PU にはローカルメモリとキャッシュメモリが備えられていること想定している。

MILL の主要な処理を以下に述べる。

**プロセッサ部:** コンパイル及びリンケージされた、ターゲット・アプリケーションプログラムを読み込み、プログラムカウンタの指す番地からインストラクションの実行をシミュレートする。

**キャッシュ部:** 共有メモリに対するアクセスが生じた時に呼ばれ、ブロックの read/write を行う。hit/miss の判定後、指定されたキャッシュプロトコルに従って無効化やブロック転送を行い、共有メモリや他プロセッサが持つキャッシュとの一貫性を保つ。

**バス・共有メモリ部:** バスリクエストが生じた時に呼ばれ、IEEE の Futurebus プロトコルに従ったバスオペレーションをシミュレートする。バスリクエストの内容によっては共有メモリの read/write を行い、バス待ち時間や占有時間などの情報採取も行なう。

## 3 時刻管理機構

並列化 MILL の時刻管理機構には、シミュレーションの対象となる要素が独自に時刻を管理する分散時刻管理機構の適用を検討した。

分散時刻管理機構の方法としては、コンサーバティブ法 [3] とバーチャルタイム法 [4] が広く知られている。しかしながらバーチャルタイム法ではインストラクション毎の履歴が必要であり、その情報を退避するメモリ等の資源消費が問題となる。

コンサーバティブ法は、そのデッドロック回避方法である Null-message による方法 [3] や間合せに基づく方法 [5] と併せて用いられる。共にメッセージ送受信を基本としたアルゴリズムだが、Null-message はデッドロックが生じる可能性に関係なく一定時刻毎にメッセージ送信が必要でありメッセージ処理によるオーバヘッドが懸念される。間合せに基づく方法においては、間合せの対象となる論理プロセスの状態を共有メモリを介して参照する方法 [2] が提案されており、その並列処理における有効性が確認されている。

そこで、MILL の並列処理ではこの間合せを共有変数の参照で代替する方法を参考にし、効率的な時刻管理処理の実現を図ることとした。

## 4 並列計算機 ATTEMPT-0

並列化 MILL を実装する並列計算機はローカルメモリを持つ共有メモリ型並列計算機 ATTEMPT-0 [6] である。最大 20 枚のプロセッサボードが Futurebus(IEEE P896.1) によって接続可能で、バスを

\*An implementation of a multiprocessor simulator MILL on a multiprocessor

<sup>†</sup>Hiroshi ENDOH, Hitachi, Ltd. Government & Public Corporation Information Systems Division

<sup>‡</sup>Takuya TERASAWA, Tokyo Engineering University

<sup>‡</sup>Hiroyuki KITOH, Dai HATTORI, Junji YAMAMOTO, Hideharu AMANO, Keio University

介して最大 16MB の共有メモリにアクセスできる。各プロセッサボードには、CPU に MC68030、FPU に MC68882 を用いており、共有メモリに対する 64KB のキャッシュメモリと 4MB のローカルメモリを持っている。また、プロセッサ間の同期機構として一種のブロードキャストメモリである Synchronizer という専用ハードウェアが装備されている。

## 5 実装

ATTEMPT-0 への実装は軽量なスレッド環境を提供する EULASH [7] カーネルを実行基盤として用いた。

並列実行の単位は対象とする並列計算機のプロセッシング・ユニット (PU) である。この並列処理単位には PU を構成するプロセッサ、キャッシュなどの機能と時刻管理機構が盛り込まれる。バス・共有メモリ部は、プロセッサを一台占有させて PU 処理とは別個に行なう。

この PU 処理は EULASH におけるスレッドとして実行される。スレッドを、シミュレートするプロセッサ台数分作成した後、ATTEMPT-0 の利用可能なプロセッサに対して可能な限り均等に配置する。プロセッサ内の各スレッドに対しては、ラウンドロビンで順次スケジューリングする。

時刻管理機構における問合せ処理は、共有メモリやバス等の共有資源に対する要求発行時に行われる。問合せの成否は、自身の現在時刻が他 PU の時刻と比較して最小であるときに成功となり、それ以外は失敗となる。成功した場合は共有資源に対する処理を進め、失敗した場合はコンテキスト・スイッチを働かせて実行スレッドを切り替える仕様とした。また、各スレッドから参照される問合せ対象の PU 状態変数は、Synchronizer に配置して高速アクセスを図っている。状態変数のような頻りに参照と更新が行なわれるデータは、ATTEMPT-0 のキャッシュプロトコルがライトスルーであるため、共有メモリよりも Synchronizer に配置した方が高速アクセスが望める。

## 6 評価

評価に用いたターゲットプログラムは、ヤコビ法による連立一次方程式の解法プログラムである。シミュレート対象とするプロセッサ数は 10 プロセッサとし、その際の MILL の並列処理における実行速度の台数効果を測定した。

台数効果は得られており、速度が 8PU で約 2.4 倍向上している。5PU から伸びが鈍化している理由は、スレッドの本数と実行プロセッサの台数が等しくなってきたために、問合せ成功までの待ち時間が別スレッドの実行に充てられなくなったことと、スレッドの負荷分散が均等でないために遅いプロセッサの影響を受けたためと推測される。

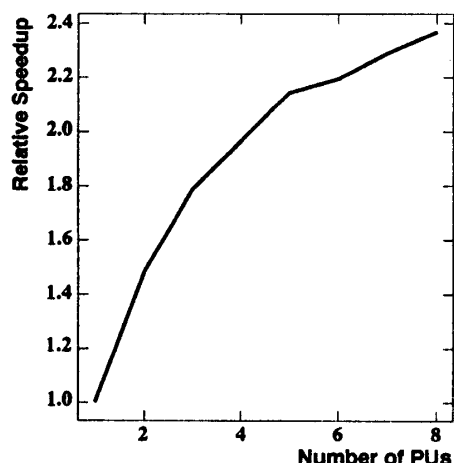


図 1: 実行速度

## 7 まとめ

本研究では、MILL を高速化するために並列化し、ATTEMPT-0 上で実装と評価を行なった。その結果、シミュレーション時間は実行 PU 台数の増加と共に高速化していることが確かめられた。

今後は他の時刻管理手法も考慮し MILL に最適なアルゴリズムの比較検討を行なう予定である。

## 参考文献

- [1] T.Terasawa, H.Amano, "Performance Evaluation of the Mixed-protocol Caches with Instruction Level Multiprocessor Simulator" Proc. of IASTED International Conference on Modeling and Simulation MS'94, May 1994
- [2] 工藤, 木村, 寺澤, 天野, "問合せに基づく並列論理シミュレーションアルゴリズム" 電子情報通信学会論文誌 Vol J75-D-1, No4, pp.211-231, Apr.1992
- [3] Chandy K.M, and Misra J., "Distributed Simulation: A case study in design and verification of distributed programs", IEEE Trans. Software Eng. 5, 5, pp.440-452, Sept.1979
- [4] Jefferson D.R., "Virtual time", ACM Trans. Program. Lang. & Syst., 7, 3, pp.365-372, Jul.1985
- [5] Chandy K.M, and Misra J., "Distributed deadlock detection", ACM Trans. Comput. Syst., 1, 2, pp.144-156, May 1983
- [6] 鳥居, 竹本, 天野, 小椋, "バス結合型並列計算機の交信用メモリの性能評価", 情処論文誌 Vol.33 No.3, 1992
- [7] 山本, 服部, 徳吉, 大和, 天野, "Local メモリを持つ共有メモリ型並列計算機を効率的に使用するための環境 EULASH", SWoPP94, 情処研報, 94-OS-65, pp.17-24, Jul.1994