

並列計算機アドレステーサ MILL-lite の PVM 上への実装

2J-5

上田 広満† 神田 和仁† 寺澤 卓也† 工藤 知宏†

東京工科大学情報工学科†
東京工科大学情報通信工学科†

1 はじめに

我々は Mill-lite と呼ぶ、シミュレーションにより並列計算機のアドレステーサを得る並列計算機アドレステーサを開発している。

MILL-lite は、単一の WS 上で実行すると、多くのプロセッサからなる並列計算機のアドレステーサを得るには膨大な実行時間が必要である。そこで、PVM[1]を用いた並列処理により計算処理の高速化を図っている。PVM(Parallel Virtual Machine) は、オークリッジ国際研究所 (ORNL) で開発されたソフトウェアで、ネットワークで接続されたコンピュータを、仮想的に分散メモリ型並列計算機として扱うことを可能にするシステムである。

2 アドレステーサ MILL-lite

2.1 MILL-lite の構成

アドレステーサ MILL-lite は、共有メモリ型並列計算機の評価用シミュレータ MILL [2][3] を基に、プロセッサと共有メモリ部だけをとりだし、高速なアドレステーサ生成を目指したものである。

図 1 に MILL-lite の構成図を示す。プロセッサ部はモトローラ社の MC680X0 を命令レベルでシミュレートし、アドレステーサを生成する。

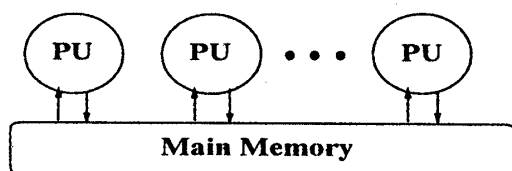


図 1: アドレステーサ Mill-lite 構成図

Address Tracer Mill-lite and its implementation on PVM
Hiromitsu UEDA, Kazuhito KANDA, Takuya TERA-
SAWA, Tomohiro KUDOH, Tokyo Engineering University

2.2 MILL-lite のメモリ管理

共有メモリ型並列計算機上で並列に動作するプログラムを実際に動かす場合、同一のテキストを複数のプロセッサで実行するモデルが考えられる。この際には、テキストは共有されるため、変数の論理アドレスは全プロセスで同一になる。しかし、実際にはプロセス間で真に共有される変数 (共有変数) とプロセス毎の変数 (ローカル変数) があり、異なるプロセスが同一の論理アドレスをアクセスしても、MMU により共有変数は同一の、ローカル変数は異なる物理アドレスへの割り付けが行なわれる。

MILL-lite では、MMU に相当する操作を簡単にするために、アドレス空間をテキスト、データ、ヒープ、共有変数の 4 組の領域に分け、それぞれプロセッサ毎に論理-物理アドレス変換を行なった。テキストは不変であるから、テキストと共有変数領域へのアクセスはプロセスが異なっても同一のアドレスに変換され、データ、ヒープは異なるアドレスに変換される。このアドレス変換はプロセッサ毎に行なっているため、1 プロセッサにつき 1 プロセスのトレースしか採取できない。

また、fork 操作をサポートしており、この操作ではデータとヒープ領域のコピーがつけられて新たなプロセッサ上でプロセスの実行が開始される。

3 PVM 上への移植

3.1 設計方針

MILL-lite を PVM 上へ実装するには、メインメモリに対する多くのメモリアクセスをどのように処理するかという問題を解決しなくてはならない。メインメモリアクセスの処理が全体の処理性能を制限するボトルネックとならないように慎重に設計する必要がある。また、複数の実プロセッサ上でシミュレーションが行なわれることになるので、時刻管理の問題も考えられ

る。そこで、PVMにおける通信のオーバーヘッドを考慮して、できるだけオーバーヘッドの小さなシステムを構築することを目指す。

3.2 設計

基本的には、MILL-liteを共有メモリ部とプロセッサ部に分け、この間で通信を行なう構成をとる。図2にMILL-liteのPVM上への実装の、全体構成の一例を示す。

このままでは、PVMでの交信がボトルネックとなるので、以下のような手法により交信量を削減する。

1. テキスト領域は不変であるので、各プロセッサ上にコピーを持ち、テキスト領域へのアクセスが交信を発生しないようにする。
2. データ、ヒープはローカル変数の領域なので、これらの領域も各プロセッサ上におくことによりアクセスが交信を発生しないようにする。
3. 共有メモリを複数のPVMのタスクに分割することにより、交信が分散するようにする。
4. 各プロセッサにキャッシュを置くことにより、交信量を減らす。この場合、コヒーレンシ維持手法を考慮する必要がある。

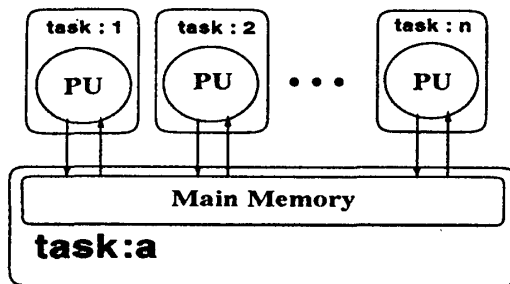


図2: Composition of Mill-lite on PVM

3.3 実行環境

最終的には、101台のS-4/LX (SUN Sparc Station LX) ワークステーション (WS) が図3のようにネットワークで相互に接続され、ファイルサーバでユーザファイルを共有している環境での実行を目標としている。

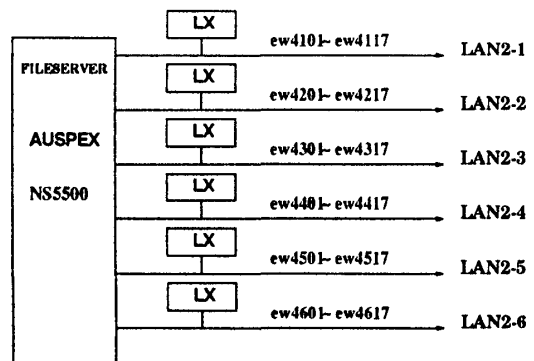


図3: 実行環境

4 おわりに

設計の頁で挙げた交信量削減手法を用いない場合、1台のプロセッサと1つのメモリモジュールをそれぞれ1つのPVMタスクに割り当てれば、PVMを用いない場合に比べて約8倍の処理時間を要することがわかった。現在、各交信量削減手法をとり入れた、高速にアドレステレースを生成できるシステムを作成中である。

参考文献

- [1] Al Geist, Adam Beguelin, Jack Dongarra, Robert Manchek, Vaidy Sunderam: "PVM3 USER'S GUIDE AND REFERENCE MANUAL", 1994.
- [2] 寺沢卓也, 天野英晴: "バス結合型並列計算機のプロセッサ間交信評価システム", 情処研報.ARC-89-25 1991.
- [3] 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.