

SMP 型計算機を活用する軽量プロセス・ライブラリ

2D-5

—スレッド間同期機構の実現と評価—

田村友彦 森本浩通 海江田章裕 小熊寿 鈴木貢 中山泰一

電気通信大学 情報工学科

1 はじめに

近年、マルチプロセッサ・システムの商用化により、複数のプロセッサを持つ並列計算機がより身近なものとなってきた。それに伴い、SMP 型 (Symmetric Multi Processor) 計算機と呼ばれるマルチプロセッサ・システムに対応した UNIX が提供されるようになった。

SMP 型計算機においてプロセッサを有効に活用するためには、スレッドあるいは軽量プロセスと呼ばれるより細かい実行単位に分けて、それらを並列に実行するための機構が必要とされている。そこで我々は、SMP 型計算機に対応したスレッド・ライブラリの実現を行っている [2]。

同期機構はスレッド・ライブラリの重要な機能であり、その性能を左右する。本稿では、同期機構の実現と評価に焦点を当てて報告する。

2 スレッド・ライブラリの実現法

我々のシステムでは、移植性を高めるため、スレッド・ライブラリを C 言語で記述し、UNIX が標準で提供する機能のみを用いてユーザ・レベルで実現している。

本システムの構成を図 1 に示す。本システムでは SMP 型計算機の複数あるプロセッサのそれぞれにスレッドを割り当て並列に実行させるために、プロセッサ数の UNIX プロセスを仮想プロセッサとして生成する。

Light-weight Process Library on SMP Computers — Implementation and Evaluation of Synchronization Primitives —

Tomohiko Tamura, Hiroyuki Morimoto, Akihiro Kaieda, Hisashi Oguma, Mitsugu Suzuki and Yasuichi Nakayama

Department of Computer Science, The University of Electro-Communications

tamura-t@igo.cs.uec.ac.jp

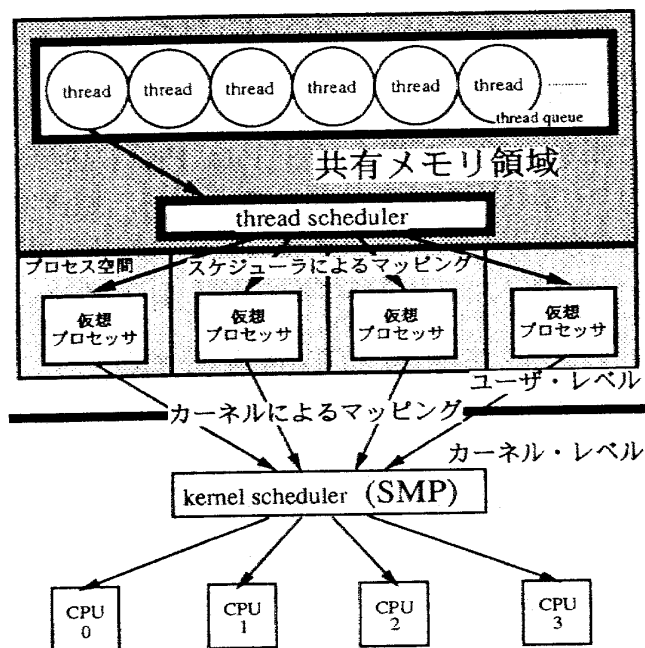


図 1: システムの構成

この場合、実行されるスレッドのコンテキストを全ての仮想プロセッサからアクセスできる領域 (共有メモリ領域) に置かなければならない。なぜなら、スケジューリング後のスレッドは、中断前と同じ仮想プロセッサで再実行されるとは限らないからである。我々は巨大な共有メモリ領域を実現するためにメモリマップド・ファイルを採用している [2]。

3 同期機構の実現

同期変数も仮想プロセッサ間で共有する必要があるため、スレッドのコンテキストと同様、図 1 の共有メモリ領域に置かなければならない。

スレッド・ライブラリの基本的な相互排除ロックに、Mutex ロックがある。ユーザ・レベルで実現したスレッド・ライブラリに PTL [1] があるが、int 型の大域変数を用いて、完全に

ユーザ・レベルで Mutex ロックを実現している。しかし、本システムでは並列に動作する仮想プロセッサ間で、Mutex ロックに対して排他的アクセスを行わなければならないので、カーネルのロック機構 (セマフォ) を用いている。そのため、ユーザ・レベルで実現した場合と比べて性能の低下が懸念される。

4 性能評価

測定には、2 個の Pentium Pro(150MHz)、64M バイトのメモリから成る SMP 型計算機と、OS に SMP 型計算機に対応した FreeBSD 3.0 を用いた。比較対象は PTL[1] である。

4.1 同期変数制御の実行時間

表 1 にスレッド間の競合が発生しない場合の Mutex 変数の制御にかかる時間を示す。PTL との処理時間の差はセマフォ操作のためのシステム・コールにかかる時間によるものと考えられる。

表 1: 同期変数制御のコスト (マイクロ秒)

	mutex_lock	mutex_unlock
PTL	36	40
本システム	58	58

4.2 アプリケーションでの性能測定

同期制御を必要とするアプリケーションとして、完全結合グラフの巡回セールスマン問題を解くプログラムを用い、性能を評価した。図 2 は、都市数を変化させたときの、PTL との実行時間の比較である。なお、縦軸は対数目盛である。

都市数が小さいときは、PTL、本システム共にスレッド未使用に比べて実行時間が長い。これは、スレッドを用いることによるオーバーヘッドのためである。本システムと PTL ではオーバーヘッドは同程度であるといえる。

都市数が大きい場合、PTL ではスレッド未使用の場合とほぼ同じ実行時間である。それに対し、本システムではスレッドの並列実行に

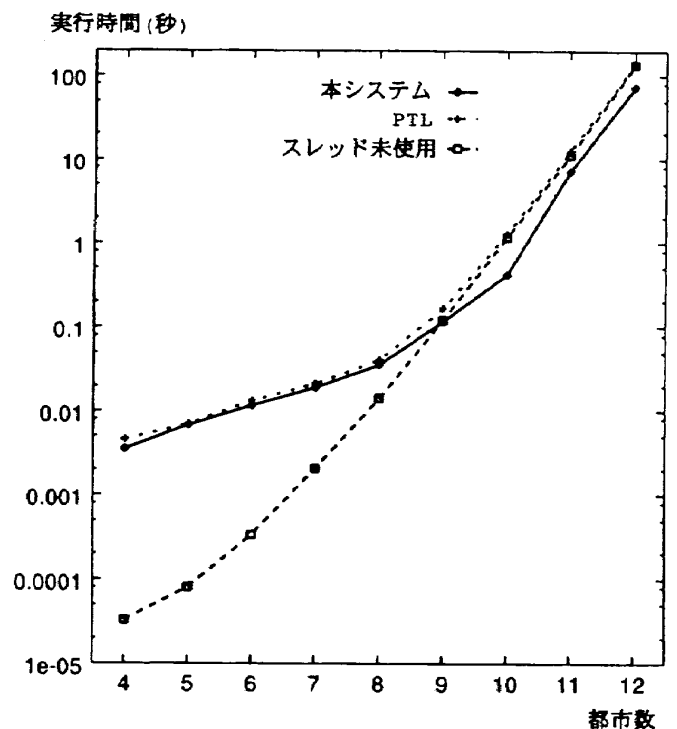


図 2: 本システムと PTL との実行時間の比較

より実行時間が短縮した。都市数が 11 のときには、PTL に対して約 60% の実行時間である。

5 おわりに

SMP 型計算機に対応するスレッド・ライブラリと、その同期機構について報告した。そして、アプリケーションを用いた性能測定から、CPU 台数に見合った性能向上を確認した。

現在、OS やプロセッサの特殊な機能に依存せず、かつ、カーネル・ロックを用いるよりも高速な同期機構の実現法を検討中である。また、より頻繁に同期が生じるアプリケーションでの性能評価を行なっている [3]。

参考文献

- [1] 安倍ほか: BSD UNIX 上での移植性に優れた軽量プロセス機構の実現, 情報処理学会論文誌, Vol.36, No.2, pp.296-303(1995).
- [2] 小熊ほか: SMP 型計算機を活用する軽量プロセス・ライブラリ, 情報処理学会プログラミング研究会報告, 97-PRO-16-3(1997).
- [3] 鈴鹿ほか: MPEG 再生のマルチスレッド化による高速化, 電子情報通信学会総合大会 (1998).