マルチコアプロセッサ上の並列プログラムの実行時間モデルに関する検討

指導教員：市川 周一　　学籍番号：113728 堂田貴裕

1 はじめに

今日のプロセッサではマルチコア・マルチスレッド化が進んでいるため，プログラムもそれに対応させる必要がある．マルチスレッド化により計算時間の短縮が期待できるが，並列度が高くなるとオーバーヘッドも大きくなる．得られる性能は問題サイズ・並列度・使用するライブラリ等に依存するため，最適な並列度を求めることは容易でない．

本研究では，最適な並列度を求め実行時間を最小化するために実行時間予測モデルを作成した．本予測モデルでは，キャッシュ溢れやメモリバンド幅を考慮している．

2 測定環境

本研究の目標は一般の応用プログラムで予測モデルを作成することである．しかし今回は，問題を単純化するため n 次元のノルム $\|\vec{x}\|^2$ を計算するプログラムを用いた．プログラムをマルチスレッド化する方法として，共有メモリ型の 2 つの API (POSIX スレッド [1] と OpenMP[2]) を使用した．

測定には Intel Core i7-860 2.8GHz，主記憶は 8GB (PC3-10700) を使用した．Core i7-860 は 4 つの物理コアを持ち，HTT (Hyper-Threading Technology) で 5 スレッド以上の同時処理を実現している．物理コア 4 つにはそれぞれ 32 KB の 1 次キャッシュと 256 KB の 2 次キャッシュがあり，すべてのコア共有で 8 MB の 3 次キャッシュがある．OS は ubuntu12.04，コンパイラは gcc 4.6.3 を使用し，POSIX スレッドライブラリには NTPL2.45 を用いた．OpenMP のライブラリは gcc に含まれている．コンパイル時の最適化オプションとして “-O3 ” を付与している．

3 測定結果・モデル化

前節で述べた 2 つの方法でマルチスレッド化を行い，実行時間のモデル化を行う．手法は共通なので本稿では POSIX スレッドについてだけ述べる．

マルチスレッド化したプログラムではスレッドを create〜join するためのオーバーヘッド時間が存在する．オーバーヘッド時間は，式 (1) により予測モデル化した．予測モデル $\theta_{pth2}(s)$ では，スレッド数 (s) が物理コア数 4 を超えるか否かで区分を行っている．式 (1) の各パラメータは実測値から最小二乗法を用い抽出する．

$$\theta_{pth2}(s) = \begin{cases} \mu_{20} \cdot s + \nu_{20} & (1 \le s \le 4) \\ \mu_{21} \cdot s + \nu_{21} & (5 \le s \le 16) \end{cases} \quad (1)$$

プログラムの逐次実行時間は，式 (2) により予測モデル化した．double 型の各要素は 8 B，2 次キャッシュ容量が 256 KB なので，$n \ge 2^{15}$ で 2 次キャッシュ溢れが生じる．同様に 3 次キャッシュ容量は 8 MB なので，$n \ge 2^{20}$ で 3 次キャッシュ溢れが生じる．そこで予測モデル $T_3(n)$ は，キャッシュ溢れが生じる n でモデル式を切り替えている．

$$T_3(n) = \begin{cases} \alpha_{30} \cdot n + \beta_{30} & (n < 2^{15}) \\ \alpha_{31} \cdot n + \beta_{31} & (2^{15} \le n < 2^{20}) \\ \alpha_{32} \cdot n + \beta_{32} & (2^{20} \le n) \end{cases} \quad (2)$$

計算時間はスレッド数 (s) に依存するため，実行時間は式 (3) によりモデル化できると考えられる．

$$T_{pth1}(n,s) = \theta_{pth2}(s) + T_3(n)/s \quad (3)$$

図 1 に，予測モデル $T_{pth1}(n,s)$ による予測実行時間と実測値 $T_{pth0}(n,s)$ との誤差を示す．$n \ge 2^{20}$ で誤差が大きくなっているのは，3 次キャッシュが溢れてデータ転送速度が主記憶のメモリバンド幅により制約されるためと考えられる．

そこでメモリバンド幅制約を考慮した予測モデル $T_{pth2}(n,s)$ を提案する．この予測モデル $T_{pth2}(n,s)$ は $n \ge 2^{20}$ でモデル式を切り替える区分関数である．3 次キャッシュが溢れた後は，スレッド数に関わらず，演算時間はデータ転送量の一次関数としている．式 (4) の各パラメータは実測値 $T_{pth0}(n,s)$ から最小二乗法を用い抽出する．

$$T_{pth2}(n,s) = \begin{cases} \theta_{pth2}(s) + T_3(n)/s & (n < 2^{20}) \\ \theta_{pth2}(s) + \alpha_4 \cdot n + \beta_4 & (2^{20} \le n) \end{cases} \quad (4)$$

予測モデル $T_{pth2}(n,s)$ で予測した実行時間と実測値 $T_{pth0}(n,s)$ との誤差を図 2 に示す．予測モデル $T_{pth2}(n,s)$ を用いることで $n \ge 2^{20}$ での実測値との乖離がなくなり，最大誤差は 19%まで改善された．

4 おわりに

本研究では予測モデルのモデル式を切り替えることでよりモデルの精度を改善することができた．予測モデル $T_{pth1}(n,s)$ では最大誤差が 65%であったが，予測モデル $T_{pth2}(n,s)$ で 3 次キャッシュが溢れた場合のメモリバンド幅を考慮することで，最大誤差が 19%まで改善することができた．

OpenMP を用いた場合のプログラムにおいても同様の手法でモデル化を行った結果，並列度 8 の $n \le 2^{11}$ を除き 20%以下の誤差で実行時間を予測することができた．

現在はベンチマークプログラム HPL (High Performance Linpack) を用い単一プロセッサ上で測定とモデル化を行っている，今後は複数の計算機を使用した PC クラスタ上で各種ベンチマークを実装し，性能を予測するモデルを作成していく．

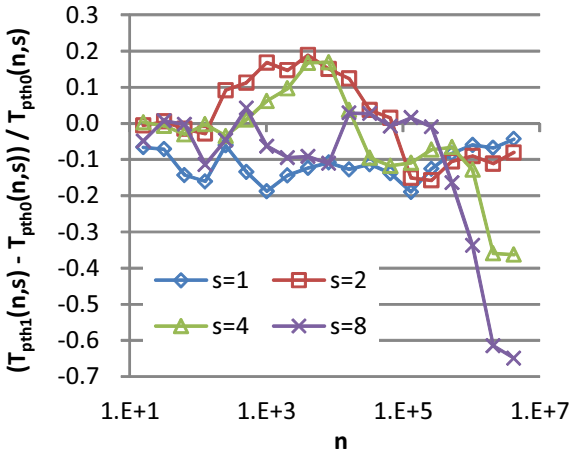

図 1: 予測モデル $T_{pth1}(n,s)$ と実測値との誤差

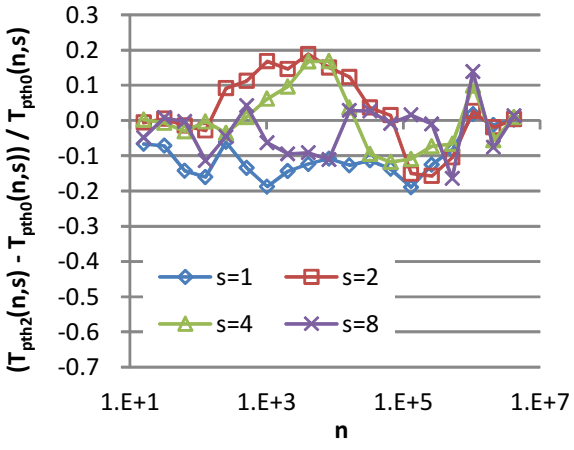

図 2: 予測モデル $T_{pth2}(n,s)$ と実測値との誤差

参考文献

[1] *pthreds (7)*, Linux Programmer’s Manual, Section 7.

[2] the OpenMP Architecture Review Board. *OpenMP*, 2012.