

4 X - 8

MPI を用いた データ並列 C 言語 NCX の実装

河内 隆仁[†], 渡邊 誠也[†], 小宮 常康[†], 湯浅太一[†][†]京都大学大学院工学研究科情報工学専攻

1 NCX について

NCX[1, 2] は、データ並列計算のための拡張 C 言語である。仮想プロセッサ (virtual processor, 以下 VP と略す) という概念を導入することによって言語の並列化を行っている。言語セマンティクス上は SIMD (Single-Instruction, Multiple-Data) 型を採用している。

2 MPI 版 NCX の実装

NCX の処理系は NCX で書かれたソースプログラムを C 言語に変換し、C コンパイラによってコンパイルした後、NCX 標準ライブラリとランタイムライブラリをリンクして実行形式を得ている。

本実装では、以下のような流れで NCX のソースを処理する。

1. NCX に共通するプリプロセッサとしてすでに実現されている NICS (NCX Intermediate Code Synthesizer)[3] を利用し、ターゲットに依存しない中間コードを得る。NICS は、仮想プロセッサ間で同期が必要なポイントを検出する。
 2. トランスレータにより、中間コードから MPI[4] のライブラリ関数および NCX のライブラリ関数呼び出しを含む C コードに変換する。
 3. C コードから実行形式への変換は、C コンパイラを用い、MPI および NCX のライブラリをリンクすることによって行う。
2. の段階で、MPI のライブラリ関数を呼び出すのは、NCX における仮想プロセッサ間の変数の参照・代入と、リダクション演算を行う場面である。

また、MPI による実装のため、以下のような仕様を持っている。

1. 利用するプロセッサ数を実行形式を起動する時に決定できる。
2. VP の数が、利用できるプロセッサ数よりも多いときには、ループを使って複数の VP の実行を 1 プロセッサで担当する。この際、割り当ては単純に領域をプロセッサ台数分にブロック分割する。
3. VP 上の変数は、その VP を担当するプロセッサが受け持つが、VP の数がプロセッサ数よりも多いときには、そのプロセッサが担当する VP の個数分の配列として持つ。アクティブな VP の切り替わり時には変数用の配列領域の割り当てや解除をランタイムライブラリが行う。

このトランスレータでは、仮想プロセッサ間の通信パターンを解析し、以下の場合には通信パターンに応じた通信コードを出力する。

- すべての仮想プロセッサが同一の仮想プロセッサの値を参照している場合
ブロードキャスト型の通信であるので、ブロードキャストのコードを出力する。
- 近傍通信を行う場合
参照先がコンパイル時に近傍通信のパターンであると決定できる場合、その 2 点間での通信を行うコードを生成する。

通信が上記以外の不規則な通信である場合には、送信要求と返答のための 2 回のプロセス間通信を行うコードを出力する。

3 性能の評価

本処理系を用いて、いくつかのベンチマークプログラムを並列計算機 Cenju-3[5] 上で実行した結果を以下に示す。グラフの横軸はプロセッサ数、縦軸は手書きの逐次版 C プログラムをプロセッサ 1 台

An MPI-Based Implementation of Data-Parallel C Language NCX

Takayoshi KOUCHI[†], Nobuya WATANABE[†], Tsuneyasu KOMIYA[†], Taiichi YUASA[†]

[†]Graduate School of Engineering, Kyoto University

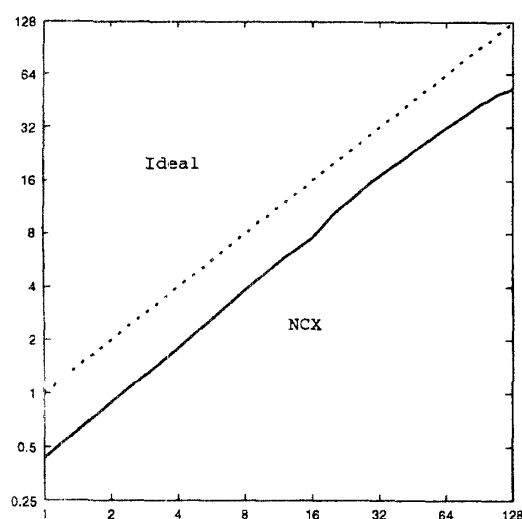


図 1: pi.ncx の高速化

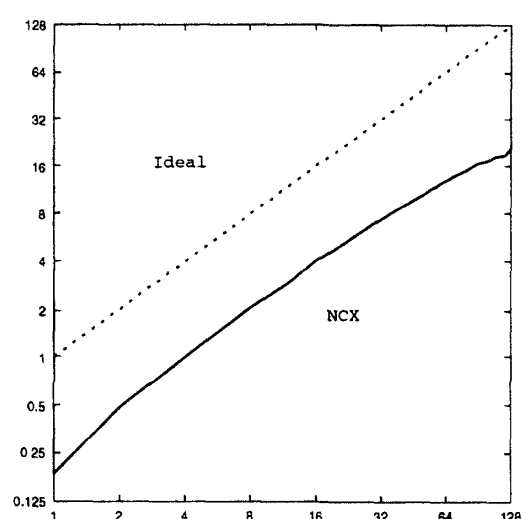


図 2: jacobi.ncx の高速化

で実行したときに対する速度比を表している。

1. 円周率 (図 1)

数値積分を用いた計算を行い、通信は計算のループ中に行ない、通信は最後のリダクション演算のみであるため、良好な並列化の効果が表れている。

2. 4 近傍の平均による温度分布 (図 2)

ループ内での通信を行っているが、並列化の効果は出ている。これは、通信が近傍通信であるため、通信コードの最適化によって、通信回数を削減できているためである。台数が 4 台以下で、台数倍以上の高速化が起こっている。これはメモリのキャッシュヒット率が原因であると考えられる。

4 まとめ

今回の実装によって、原理的には MPI の動く全ての計算機環境で NCX が利用可能となった。しかし、性能に関して、まだ十分とは言えない状況であり、今後は、様々な最適化を行えるよう、コンパイラのチューニングを行わねばならない。

参考文献

- [1] 湯浅 太一, 貴島 寿郎, and 小西 浩. データ並列計算のための拡張 c 言語 ncx. 電子情報通信学会論文誌, J78-D-I(2):200-209, February 1995.

- [2] 文部省重点領域研究「超並列原理に基づく情報処理基本体系」B 班. 超並列 C 言語 NCX 仕様書 version 3, October 1993.

- [3] Taiichi Yuasa. *NICS Output Format*. Toyohashi University of Technology, February 1996.

- [4] MPI Forum. *Mpi: A message-passing interface standard*, June 1995.

- [5] 丸山 勉, 加納健, 広瀬哲也, 中田登志之, 松村一弘, 浅野由裕, and 稲村雄. 並列コンピュータ cenju-3 のアーキテクチャとその評価. 電子情報通信学会論文誌, J78-D-I(2):59-67, February 1995.