

## 超並列 C 言語 NCX における式の遠隔評価

2 P-6

日高 宗一郎, 川合 史朗, 相田 仁, 齊藤 忠夫

東京大学 工学部

## 1 はじめに

超並列 C 言語 NCX は、データ並列計算のために設計された拡張 C 言語である [1]。我々は NCX で記述された並列プログラムを MIMD/分散メモリ型計算機上で実行するためのコンパイラの開発に取り組んでいる。NCX では仮想プロセッサ (以下 VP) 間の通信が遠隔参照という非明示的な形で記述されるが、ターゲットマシン上では Dataparallel C [2] (以下 DPC) 処理系における通信ライブラリ関数を用いて実現している。NCX の並列構文の実行において、式の評価をどのプロセッサが行なうかという選択は、このプロセッサ間通信の回数やコストを左右し、効率に与える影響が大きい。DPC や現状の実装においては、全ての間接参照を別々にローカルプロセッサにコピーする方針を採用している。

本稿では、式の階層構造間のデータフローに基づいてプロセッサ間通信を抽出し、遠隔評価により無駄な通信を減少させる方針を提案し、その有効性を示す。

## 2 NCX における式の評価と VP 間通信

NCX において式文の式を評価するように定義されている VP をここでは Local VP (以下 LP) と呼ぶ。言語仕様上は任意の部分式は LP から可視である。但し、式全体の値が LP に渡りさえすれば、必ずしも全ての部分式の値が LP に知れなくてもよい。例えば加算式を評価する場合、二つの被演算数の値は、実際に加算を行なう VP に渡せばよく、必ずしも LP に渡す必要はない。ここでどの VP に加算を行なわせるかの選択の余地が生じる。また、NCX にはポインタ式の値がどの VP のアドレスを表すかを特定するためにホームプロセッサ (Home Processor: 以下 HP) の概念が導入されている。間接参照や演算は、それぞれ被演算数の HP 上で行なわなければならない。この場合は遠隔評価が必要となる。本稿ではこの HP の概念を拡張し、オブジェクトを参照する式に対しても、それを所有する VP を HP と呼ぶ。VP 間通信は、式の各階層間において被演算数を持つ VP から演算を行なう VP へ向かって生じると

考えることが出来る。VP 間通信のパターンは、変数の遠隔参照を示す識別子の後のフィールドインデックスのパターンにより表される。グローバルな視点からは変数アクセスのパターンはアクティブな VP 集合から HP の特定に使われた識別子の属するフィールドの VP 全体の集合の中への写像である。この写像が一般に実行時にしか分からない場合は次節で示すランダム通信として処理される。

## 3 ランダム通信の実現方式とフェーズの概念

我々が現在の NCX 実装で採用している  $x=y@(\text{expr})$  のような遠隔参照におけるランダム通信は、DPC の通信ライブラリ関数の仕様を参考にしている。その内、任意の物理プロセッサ (以下 PP) 上の任意のアドレスのデータを任意の PP 上の任意のアドレスにコピーする `random_read` 関数は以下のように実現されている。

- (1) コピー元、PP 番号、アドレス、コピー先の PP 番号、アドレスを格納したバケットの配列が引数として渡される。
- (2) (1) のバケットをコピー元の PP へ配送する。
- (3) (2) の配送で受けとったバケットのコピー元のアドレスにあるデータ、コピー先の PP 番号、アドレスを格納したバケットを生成する。
- (4) (3) で作ったバケットの配列を受けとり、バケットのデータを指定の PP 上のアドレスへ書き込む通信ライブラリ関数 (`random_write`) を呼ぶ。

ここで述べた `random_read` 関数は、任意の第三者通信を実現するが、NCX における遠隔参照では、コピー先に呼び出し元の PP 番号を指定して結果を呼び出し元に書き戻させている。上述の (3) の段階では LP ではなくデータの持ち主の動作をしていることに着目する。

一般に、VP の計算、記憶の機能は有限の PP によりそれぞれループ、配列で実現されるが、通常の実行モードでは LP をエミュレートしている。この場合アクティブな VP のイテレーションのみが選択的に実行され、PP がエミュレートする VP の数まで繰り返される。これに対し、(3) のようなモードでは LP から送られてきたリクエスト (ここではデータ転送依頼) の処理をエ

"Remote Evaluating Expressions in Massively-Parallel C Language NCX"

Soichiro Hidaka, Shiro Kawai, Hitoshi Aida and Tadao Saito

Faculty of Engineering, The University of Tokyo

ミュレートしており、受けとったリクエストの数(ここでは random\_read パケットの数)だけ繰り返される。

本稿では、この (3) の操作をライブラリ関数から外に切り出し、そこで遠隔評価の機能をエミュレートさせる方式を提案する。エミュレートしている VP の集合が切り替わることをここで”フェーズが切り替わる”と言う。フェーズを、相対的に上位の演算をするための被演算数を計算すべき VP になりかわったモードとして対応付ければ、式の階層構造にフェーズを対応させることにより任意の式の階層間通信が効率良く実現できる。

同じ階層の被演算数に相当する式(2項演算式の被演算数等)の値を計算する HP エミュレーションループは同じフェーズに同居させることが出来る。これは複数の種類のリクエストパケットを処理することにより、同時に複数の種類の HP をエミュレートすることが可能になるからである。

フェーズは C 言語の複文のように入れ子に出来る(抜けたらもとのフェーズに戻っている)。この場合、データの行き先は上位のフェーズである。

フェーズの切り替えは、被演算数の HP が式の HP と異なる場合に起こる。切り替わる可能性のあるのは間接参照(\*や&)の場合のみ(切り替わらない場合もある)である。C 言語風のブロックの形式で表現すると以下のようになる。

```

/* フェーズ 0 */
下位のフェーズへのリクエストの配送
/* 被演算数が複数存在する式の場合は異なる種類の
リクエストを一括配送。現フェーズ上のデータが
下位のフェーズで使われる場合にはリクエスト
パケットの可変長部分にそれぞれのスロットを追加し、
そのデータを載せる */
{
  /* フェーズ 1 */
  HP エミュレーションループによる上位のフェーズからの
  リクエストの実行(この中に別のフェーズが入れ子に
  なっていても構わない)
  /* HP レベルでの同期(代入と参照の間の同期など)は
  エミュレーションループの切れ目に対応する */

  評価結果の上位のフェーズへの書き戻し
}
/* フェーズ 0 に戻っている */
    
```

#### 4 変換例

$x = *(p@i) + q@(j)++ + r@(k) + s$  の評価における評価依頼と評価結果、HP エミュレーションの流れを図 1 に示す。NCX の仕様には忠実な実装では、 $p@i$ 、 $q@j$ 、 $r@k$  それぞれに対して random\_read1 回ずつ、間接参照に関する random\_read1 回、 $q@j$  のインクリメントに関する random\_reduction1 回に変換されてい

た。図 1 の例では、間接参照式のポインタ式の HP 上でポインタ式を求める加算を行なわせ、 $q@j$ 、 $r@k$  の各値は LP にコピーせず、HP(i) に直接向かわせている。また加算の被演算数としてのインクリメント式  $q@j++$  は、LP から発行されたリクエストの数だけのインクリメントが参照と同時に PP レベルでの同期を伴わずに行なわれ、アクティブな LP の数だけのインクリメントされるという言語仕様が効率良く実現されていることが分かる。

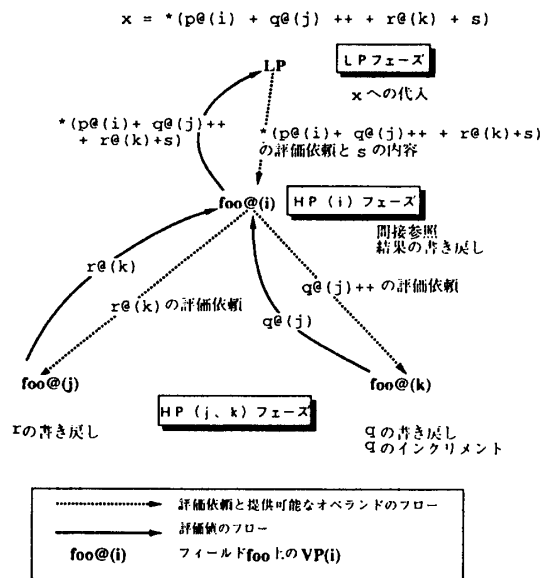


図 1: 式の階層と評価依頼/結果の流れ

#### 5 まとめ

本稿では、NCX の式の評価においてフェーズの切り替えによる HP エミュレーションの概念を導入し、ランダム通信に使われる通信ライブラリ関数の機能を拡張することにより、式の遠隔評価が従来の遠隔参照方式に比べて少ない通信で実現出来ることを示した。式の階層構造がフェーズの切り替えに直接対応付けられており、直観的なコンパイルが可能となった。

#### 参考文献

[1] 湯浅 太一, 貴島 寿郎, 小西 浩: データ並列計算のための拡張言語 NCX, 電子情報通信学会論文誌, Vol. J78-D-I, No. 2, pp. 200-209 (1995).

[2] Hatcher, P. J. and Quinn, M. J.: DATA-PARALLEL PROGRAMING ON MIMD COMPUTERS, The MIT Press (1991).