4T－7

粗粒度並列処理における Doall/シーケンシャルループ間データローカライゼーション手法

吉田 明正†　前田 誠司††　尾形 航†　山下 浩一郎†　笠原 博徳†
†早稲田大学理工学部　††(株) 東芝

1　はじめに

マルチプロセッサシステム上での Fortran プログラムの粗粒度並列処理手法としてマクロデータフロー処理 [4] [5] が提案されている．マクロデータフロー処理のように，ダイナミックスケジューリングを用いて粗粒度タスク (マクロタスク) を実行時にプロセッサに割り当てる方式では，従来，マクロタスク間で共有されるデータを集中型共有メモリ上に置き，マクロタスク間のデータ授受は集中型共有メモリを介して行なう方法がとられていた．しかし，このような方式では，集中型共有メモリを介したデータ転送オーバヘッドが大きくなるという問題が生じる．

この問題点を解決するためには，マクロタスク間データ授受のために，プロセッサ内ローカルメモリの利用が重要となる．このローカルメモリの有効利用では，現在までマクロタスク間 (ループ間) データ授受に関してほとんど研究がされておらず，単一ループ内でローカルメモリの有効利用を行なう研究として，Array Privatization 法 [1][2] が提案されているにすぎない．また，分散共有メモリマシン上でのデータ分割・配置に関しては，Anderson と Lam がプロセッサ間通信の最小化を目指した静的なデータ分割・配置法 [3] を提案している．しかし，この方法は，マクロデータフロー処理のように実行時に手順とデータを動的に配置し，粗粒度並列処理を行う方式には適用できない．

そこで，筆者等は，マクロデータフロー処理において，複数 Doall ループ間で，Doall ループの並列性を利用しつつ，ローカルメモリ経由でデータ授受を行う方法 [6] を提案してきた．この方式では，ローカルメモリ経由データ授受を行なう複数マクロタスクを，コンパイル時のタスク融合を用いて，実行時に同一プロセッサに割り当てている．このため，タスク融合のできない複数マクロタスク間 (例えば，分割された部分 Doall ループと部分シーケンシャルループの間) では，ローカルメモリを介してデータ授受を行なうことが困難であった．

本稿では，そのような問題点を解決し，Doall ループとシーケンシャルループ間でのデータローカライゼーション手法を提案する．本手法では，Doall ループとシーケンシャルループを，配列データの使用範囲が等しくなるように複数の部分 Doall ループと部分シーケンシャルループに分割し，その後，データ転送量の多い部分 Doall ループと部分シーケンシャルループを実行時に同一プロセッサに割り当てて，ローカルメモリ経由データ授受を実現する並列マシンコードを生成する方式をとる．

2　マクロデータフロー処理

本マクロデータフローコンパイレーション手法 [4] [5] では，Fortran プログラムを，BPA(擬似代入文ブロック)，RB(繰り返しブロック)，SB(サブルーチンブロック) の3種類のマクロタスク (MT) に分割する [5]．

コンパイラは次に，BPA, RB, SB 等のマクロタスク間のコントロールフローとデータフローを解析する．この解析結果は，マクロフローグラフ (MFG) [4] で表現される．

マクロタスク間には，コントロール依存とデータ依存が存在するので，本コンパイレーション手法では，コントロール依存とデータ依存を考慮した マクロタスク間並列性を最大限に引き出すために，各マクロタスクの最早実行可能条件解析 [4] を適用する．この各マクロタスクの最早実行可能条件は，マクロタスクグラフ (MTG)[4] で表すことができる．

この後，コンパイラは，マクロタスクを実行時にプロセッサクラスタ (PC) に割り当てるためのダイナミックスケジューリングコードを生成する．

3　Doall/シーケンシャルループ間データローカライゼーション

本データローカライゼーション手法では，コンパイラが，データ依存関係のある Doall ループとシーケンシャルループを，配列データの使用範囲が等しくなるように整合して部分ループに分割する．その際，それらの部分ループの内，各々が異なるプロセッサクラスタ (PC) に割り当てられると PC 間に多量のデータ転送が生じてしまう部分ループの集合は，実行時に同一 PC にスケジューリングされるようにダイナミックスケジューラに指示する．次に，コンパイラはローカルメモリを介してデータ授受を行なう並列マシンコードを生成する．以下では，このコンパイル手法を概説する．

3.1　シーケンシャルループを考慮したループ整合分割

ここでは，マクロタスクグラフ上で，データ依存エッジで接続された RB(Doall またはシーケンシャルループ) のグループ (例えば図 1(a)) が存在する場合，RB グループ内の各ループの分割により生成される部分 Doall ループと部分シーケンシャルループの間でローカルメモリ (LM) 経由データ授受を可能とするループ分割法を提案する．このシーケンシャルループを考慮したループ整合分割法は，複数 Doall ループだけを対象としたタスク融合用ループ整合分割法 [6] を拡張した方法である．以下に各手順を述べる．

1. MTG 上で単一のデータ依存エッジで接続された RB 集合を RB グループとする．
2. RB グループ内で，ある RB のイタレーションと他の RB のイタレーションとの間のデータ依存 (インター RB イタレーション間データ依存) を解析する．
3. このインター RB イタレーション間データ依存解析結果に基づいて，Doall ループとシーケンシャ

A Data-Localization Scheme among Doall and Sequential Loops for Coarse-Grain Parallel Processing,
Akimasa YOSHIDA, Seiji MAEDA, Wataru OGATA, Koichiro YAMASHITA, and Hironori KASAHARA,
Waseda University, 3-4-1 Okubo Shinjuku-ku, Tokyo 169

ルループをデータの使用範囲が等しくなるように整合分割する.

4. RB グループ内で整合分割された部分 RB 集合の中で, データ転送量の多い部分 RB 集合を, データローカライゼーショングループとする.

例えば, 図 1(a) の RB グループを, 3PC 用にシーケンシャルループを考慮したループ整合分割法で分割すると, 3 分割後の MTG は図 1(b) のようになる. また, 図 1(b) において, 各網掛け部分をデータローカライゼーション対象グループとする.

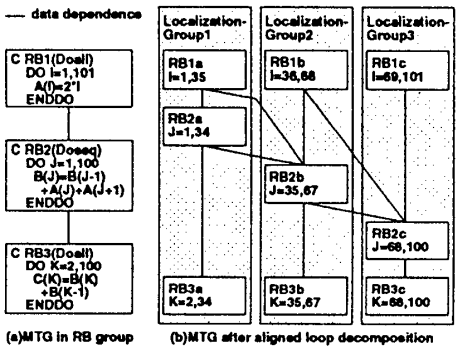

図 1: シーケンシャルループを考慮したループ整合分割の例

3.2 同一 PC 割当指定を伴うダイナミックスケジューリングルーチン生成

3.1で生成されたデータローカライゼーショングループ内の複数マクロタスク (MT) 間でデータローカライゼーションを実現するためには, これらの MT が実行時に同一の PC にダイナミックスケジューリングされていなければならない. そこで, 本手法では, Dynamic-CP 法を拡張し, コンパイラが指定した特定の MT 集合を同一の PC にダイナミックスケジューリングする方式を提案する.

本ダイナミックスケジューリング手法では, コンパイル時に, あるデータローカライゼーショングループ kにおける入口マクロタスク $MT_{k,entrance}$以外のマクロタスク $MT_{k,i}$は, $MT_{k,entrance}$と同じ PC に割り当てることを指定する.

そして実行時に, 本ダイナミックスケジューラでは, $MT_{k,entrance}$を PC に割り当てる時に, $MT_{k,i}$を割り当てるべき PC を記録し, $MT_{k,i}$は実行可能になった時, 記録された PC に割り当てられる. なお, $MT_{k,entrance}$は, その時点で各 PC の推定負荷 (その PC で実行中の MT 及び割り当ての確定している MT の推定処理時間の総和) が最小の PC に割り当てられる.

3.3 LM 経由データ授受コード生成

ここでは, データローカライゼーショングループ内の複数マクロタスク間で, ローカルメモリ (LM) 経由のデータ授受コードを生成する方法を述べる.

まず, データローカライゼーショングループ内でデータローカライゼーションを行なう場合に, LM 経由ではなく共有メモリ (CM) 経由で行なうべきデータ転送を各配列変数ごとに解析する.

次に, データローカライゼーショングループ内において, LM 経由データ転送時間が, CM 経由データ転送時間より短くなる配列変数に対して, データローカライゼーション (LM 経由データ授受) 用マシンコードを生成する.

4 OSCAR 上での性能評価

本章では, 提案手法をアプリケーションプログラムに適用し, OSCAR シミュレータ上で評価した結果について述べる.

OSCAR[7] は, ローカルメモリ (LM) と分散型共有メモリ (DSM) を持つ 32 ビット RISC プロセッサ (PE) を, 集中型共有メモリ (CM) に 3 本のバスで接続した共有メモリ型マルチプロセッサシステムである.

本性能評価に用いるスプライン補間法のプログラムは, 9 つの Doall ループ, 2 つのシーケンシャルループ (ループキャリィドデータ依存を持ったループ), 3 つの基本ブロックで構成されている.

このプログラムを 1PE 上でシーケンシャル実行すると, 処理時間は 632[ms] であった. 次に, 6PE を用いてマクロデータフロー処理で実行すると, マクロタスク間の粗粒度並列性を利用できるため, 処理時間は 188[ms] となり, 6PE 上での Doall 処理の処理時間 (218[ms]) より短縮される.

さらに, 提案するデータローカライゼーション手法を適用したマクロデータフロー処理の場合には, Doall ループとシーケンシャルループの間及び Doall ループ間で CM 経由のデータ転送オーバヘッドが軽減されるため, 6PE での処理時間は 152[ms] となり, CM 経由データ授受のマクロデータフロー処理と比べて 19.1%短縮されている.

5 まとめ

本稿では, Fortran マクロデータフロー処理において, Doall ループとシーケンシャルループの間でのデータ転送にローカルメモリを用いることにより, 負荷バランスをとりながらデータ転送オーバヘッドを軽減するコンパイル時タスクグループ割当指定付ダイナミックスケジューリングを用いた自動データローカライゼーション手法を提案した. また, 性能評価の結果より, 本手法が従来の共有メモリ経由データ授受の場合に比べて, 処理時間を顕著に短縮できることが確かめられた.

今後の課題としては, マルチグレイン並列処理におけるデータローカライゼーション手法の開発等があげられる.

本研究の一部は, 文部省科学研究費 (特別研究員奨励費 053760, 一般研究 (b)05452354, 一般研究 (c)05680284) により行なわれた.

参考文献

[1] P.Tu, D.Padua: "Automatic Array Privatization", 6th Annual Workshop on Languages and Compilers for Parallel Computing (1993).

[2] Z.Li: "Array Privatization for Parallel Execution of Loops", Proc. of the 1992 ACM Int'l Conf. on Supercomputing, pp.313-322 (1992).

[3] J.M.Anderson, M.S.Lam: "Global Optimizations for Parallelism and Locality on Scalable Parallel Machines", Proc. of the SIGPLAN '93 Conference on Programming Language Design and Implementation, pp.112-125 (1993).

[4] 本多, 岩田, 笠原: "Fortran プログラム粗粒度タスク間の並列性検出手法", 信学論 (D-I), J73-D-I, 12, pp951-960 (1990-12).

[5] 笠原, 合田, 吉田, 岡本, 本多: "Fortran マクロデータフロー処理のマクロタスク生成手法", 信学論 (D-I), J75-D-I, 8, pp.511-525 (1992-08).

[6] 吉田, 前田, 尾形, 岡本, 本多, 笠原: "マクロデータフロー処理におけるデータローカライゼーション手法", 信学技報, 93, 180, pp.81-88 (1993-08).

[7] 笠原, 成田, 橋本: "OSCAR のアーキテクチャ", 信学論 (D), J71-D, 8 (1988-08).