1J－5　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のための近細粒度タスクスケジューリング

吉田 明正†　尾形 航†　岡本 雅巳†　合田 憲人†　笠原 博徳†
†早稲田大学 理工学部

1　はじめに

マルチプロセッサシステム上でのFortranプログラムの並列処理手法として，粗粒度並列処理(マクロデータフロー処理[4])，中粒度並列処理(ループ並列化)，近細粒度並列処理を階層的に組み合わせたマルチグレイン並列処理[6]が提案されている．従来のマルチグレイン並列処理では，マクロタスク(粗粒度タスク)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も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，プロセッサ内のローカルメモリを利用したマクロタスク間データ授受が重要となる．ローカルメモリの有効利用に関しては，単一ループを対象としたArray Privatization法[1]や，複数ループ間でPE間通信の最小化を目指した静的なデータ分割・配置法[2][3]が提案されている．しかし，これらの方法は，中粒度並列処理のみを対象としているため，実行時に処理とデータを動的に配置し複数粒度で並列処理を行なうマルチグレイン並列処理には適用できない．

一方，マクロデータフロー処理においては，ローカルメモリ経由でマクロタスク間データ授受を行う方法(データローカライゼーション手法)[5]が提案されている．しかし，この方法ではマクロタスク内部を複数プロセッサ上で並列処理するマルチグレイン並列処理に適用すると，MT間で一部のデータしかローカルメモリを介して授受できないという問題点があった．

本稿では，このような問題点を解決するために，マクロタスク間データ転送を考慮した近細粒度タスクスケジューリング手法を提案する．

2　マルチグレイン並列処理

2.1　マクロデータフロー処理

マクロデータフローコンパイレーション手法[4][6]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．但し，粗粒度並列性を利用し，かつ，データローカリティを高めるため，マクロタスクの分割・融合を行い，マクロタスクの粒度の調整を行なう[4]．

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し，マクロフローグラフ(MFG)[6]を生成する．この後，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[6]を解析する．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)[6]と呼ぶ無サイクル有向グラフで表すことができる．

最後に，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．

2.2　中粒度並列処理(ループ並列化)

前述のマクロデータフロー処理によりPCに割り当てられたマクロタスクが，Doall可能RBである場合，PC内部のPEによって中粒度すなわちイタレーションレベルで並列処理される．

2.3　近細粒度並列処理

ループ並列化不可のRBのボディ部に存在する基本ブロック，及び，ループ外部のBPAでは，基本ブロック(BPA)がステートメントレベルの近細粒度タスクに分割され，PC内のPE上で並列処理される[6]．

近細粒度タスクのPC内PEへの割り当てには，データ転送を考慮したヒューリスティックアルゴリズムであるCP/DT/MISF法，DT/CP法[6]，あるいは，提案するMT間データ転送を考慮した近細粒度タスクスケジューリング手法を採用する．

3　MT間データ転送を考慮した近細粒度タスクスケジューリング

本節では，データローカライゼーションの適用されるマクロタスク間で，PE間データ転送を最小化するための，近細粒度タスクスケジューリング手法について述べる．

3.1　データ転送評価用擬似タスクの生成

本タスクスケジューリング手法では，データローカライゼーションを行なうマクロタスク集合から，データ転送評価用の擬似タスク(以後，単に擬似タスクと呼ぶ)を生成する．ここで，擬似タスクとは，MT間データ転送を考慮してMT内部の近細粒度・中粒度タスクをスケジューリングするために使用するものであり，実際にプロセッサ上で処理するものではない．ここでは，以下の3種類の擬似タスクを定義する．

(1)MTがDoallループの場合，Doallループのインデクス範囲をPC(プロセッサクラスタ)内のPE数個に均等に分割し，この分割された各部分Doallループ(中粒度タスク集合)を擬似タスクと定義する．例えば，図1(a)の部分プログラム(プログラム中からMT間データローカライゼーション適用範囲として自動抽出される)中のDoallループ*RB3*は，PCが2PEで構成される場合，図1(b)に示すように，擬似タスク*L3_1*と*L3_2*に分割される．

(2)MTが基本ブロック(BPA)の場合には，図1(b)の*BPA1*のように，基本ブロック内の各ステートメントを擬似タスクと定義する．

(3)MTがシーケンシャルループの場合には，ループ内部の各ステートメント群(ボディ部の各ステートメントをループイタレーション回数分展開したもので，ループディストリビューションにより分割されたループと似たイメージ)を擬似タスクと定義する．但し，データローカライズされるMT集合内に中粒度並列処理されるMTが含まれている場合には，擬似タス

A Near-Fine-Grain Task Scheduling Scheme for Data-Localization on Multi-Grain Parallel Processing,
Akimasa YOSHIDA, Wataru OGATA, Masami OKAMOTO, Kento AIDA, and Hironori KASAHARA,
Waseda University, 3-4-1 Okubo Shinjuku-ku, Tokyo 169

ク生成前に, シーケンシャルループのインデクス範囲を, PC 内の PE 数個に均等に分割しておく. これは, シーケンシャルループの近細粒度タスクのスケジュール (タスクのプロセッサへの割当て) をループ制御変数の値によって変更できるようにするためである, 例えば, 図 1(a) のシーケンシャルループ *RB2* は, インデクス範囲で図 1(b) の *RB2a* と *RB2b* に分割され, その後, *RB2a* と *RB2b* 内部で図 1(b) の [ ] に示した擬似タスクに分割される.

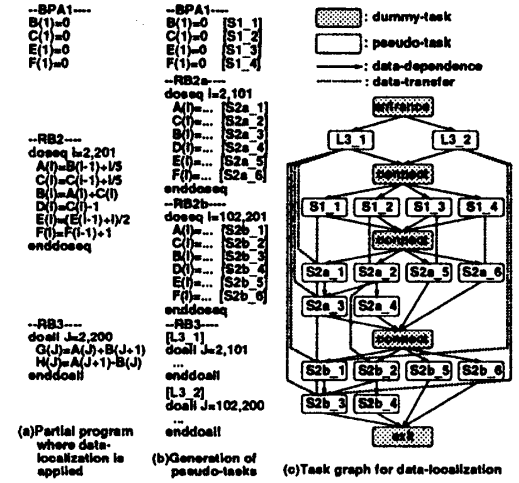

図 1: データローカライゼーション用タスクグラフの例

3.2 データローカライゼーション用タスクグラフの生成

次に, データローカライゼーションの対象となる複数 MT 間のデータ転送を表したデータローカライゼーション用タスクグラフを生成する.

具体的には, まず, 各 MT ごとに, 擬似タスク間のデータ依存 (図 1(c) において矢印付きの実線) を表したタスクグラフを作る. 次に, データローカライゼーションを行なう MT 集合中で, 中粒度並列処理の適用される MT, 近細粒度並列処理の適用される MT の順に, 各 MT のタスクグラフを接続する. このとき, 各 MT のタスクグラフの入口及び出口には, ダミーの入口ノード, 接続ノード, 出口ノード (図 1(c) では *entrance*, *connect*, *exit*) を用いる.

その後, MT 間及びシーケンシャルループ (MT) のイタレーション間において, 擬似タスク間のデータ転送エッジ (図 1(c) において太点線) を付加し, データローカライゼーション用タスクグラフを生成する. この際, データ依存エッジ及びデータ転送エッジは, データ転送時間の重み (1 データの転送時間 t_{trans} × 転送データ数) をもたせる. 例えば, 図 1(c) において, 擬似タスク *S2a_1* と *L3_1* の間では, 配列 A(2)〜A(101) のデータ転送が必要となるので, データ転送時間は $t_{trans} \times 100$ となる. なお, 前述のように, このデータローカライゼーション用タスクグラフは, データローカライゼーションのための MT 内近細粒度タスクのスケジューリングにのみ使用されるもので, MT 間の実行順序制約は表していない.

3.3 データ転送評価用擬似タスクのスケジューリング

最後に, データローカライゼーション用タスクグラフ上の擬似タスクを PC 内の PE にスケジューリングする. このとき, 近細粒度タスクスケジューリングにおけるデータ割当ての基準となる部分 Doall ループで構成された擬似タスクは, ループ制御変数の初期値の小さい部分 Doall ループから順に, PC 内の番号の小さい PE に割り当てられる. 一方, ステートメント (群) で構成された擬似タスクは, DT/CP(Data Transfer/Critical Path) 法のスケジューリングアルゴリズムにより, データローカライゼーション用タスクグラフ上の擬似タスク間データ転送時間を考慮して, PC 内の PE に割り当てる. このような擬似タスクのスケジューリングにより, 各 MT 内の擬似タスクに対応する近細粒度タスクまたは中粒度タスクのスケジュールが求められる.

4 OSCAR 上での性能評価

本性能評価では, 図 1(a) のプログラムに, 提案手法を適用し, OSCAR シミュレータ上で実行した結果について述べる. OSCAR 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 (CSM) に 3 本のバスで接続した分散/集中共有メモリ型マルチプロセッサシステムである.

図 1(a) 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 10.04[ms] であった. 次に, 4PE を用いて従来の Doall 処理により実行すると, シーケンシャルループ等を並列処理できないため, 処理時間は 8.16[ms] でありほとんど短縮されない. それに対して, 4PE からなる 1PC を用いて中粒度並列処理と近細粒度並列処理を適用すると, 処理時間は 4.36[ms] に短縮される.

さらに, 提案する近細粒度タスクスケジューリングを行ない, データローカライゼーション手法を適用した場合には, 中粒度並列処理を適用する Doall ループと近細粒度並列処理を適用するシーケンシャルループとの間で LM 経由データ授受が行われ, 1PC(4PE) 上での処理時間は 3.39[ms] に短縮された. この場合, 処理時間は CM 経由データ授受の場合に比べて 22.2%短縮されている.

5 まとめ

本稿では, マルチグレイン並列処理において, 各粒度の並列性を最大限に利用しつつ, 複数 MT 間でのデータローカライゼーションを効果的に実現するための, 近細粒度タスクスケジューリング手法を提案した. また, OSCAR シミュレータ上での性能評価の結果より, 提案手法の有効性が確かめられた.

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] M.Gupta, P.Banerjee: "Demonstration of Automatic Data Partitioning Techiniques for Parallelizing Compilers on Multicomputers", IEEE Trans. on Parallel and Ditributed System, Vol.3, No.2, pp.179-193 (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] 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), Vol.J75-D-I, No.8 (1992-08).

[5] 吉田, 前田, 尾形, 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", 情処学論, Vol.35, No.9 (1994-09).

[6] H.Kasahara, H.Honda, A.Mogi, A.Ogura, K.Fujiwara, S.Narita: "Multi-Grain Parallelizing Compilation Scheme for OSCAR", 4th Workshop on Language and Compilers for Parallel Computing (1991).