1J－7　自動並列化コンパイラにおける データプレロード・ポストストアを用いたデータ転送オーバヘッドの隠蔽*

藤本謙作, 笠原博徳†
早稲田大学理工学部‡

1　はじめに

ローカルメモリ及び分散共有メモリを持つマルチプロセッサシステム上で科学技術計算の並列処理を行なう際には、プロセッサ間データ転送オーバヘッドを最小化することが重要である。筆者等は従来より、コンパイラがローカルメモリを有効に利用し、プロセッサ間データ転送を最小とするようなデータおよびプログラムの分割およびスケジューリングを行ない、さらにその際除去できなかったデータ転送オーバヘッドを、データプレロード・ポストストア技術を用いプログラム実行とデータ転送をオーバラップさせることにより隠蔽する手法を提案している[8]。本稿では、このデータプレロード・ポストストアを用いたマルチプロセッサスーパーコンピュータ用並列化コンパイラ及びその性能評価について述べる。

2　データプレロード・ポストストアを用いたマルチプロセッサスケジューリング

本節では対象とするマルチプロセッサシステムのモデルと、データプレロード・ポストストアを用いたマルチプロセッサスケジューリングアルゴリズムについて述べる。

2.1　対象マルチプロセッサシステムのモデル

本稿では対象マルチプロセッサシステムとして、相互結合網(ネットワーク)を介して接続されたプロセッサエレメント(PE)が、それぞれローカルメモリ(LM)・分散共有メモリ(DSM)とデータ転送ユニット(DTU)を持つ分散共有メモリ型マルチプロセッサシステムを考える。ここでDTUとは他のPEのDSM上のデータへのリード・ライトを行なうデータ転送コントローラである。DTUによるデータの転送は、CPUのプログラム実行とオーバラップして行なうことができるものとする。

2.2　スケジューリング問題の定義

図1は、タスク(マクロデータフロー処理[2]のマクロタスクに相当する)間の先行制約を表したマクロタスクグラフ[2]を示している。各ノードがタスクを表し、ノード内の数字はタスク番号、左側の数字は実行時間、タスク間のエッジはデータ依存を表している。各エッジにはデータ番号とデータ転送量(転送に要する時間に換算したもの)が付加されている。但し、データ番号は複数の変数および配列中の領域の集合を指しており、対応するデータ転送量もそれらの転送に要する時間の総和である。

あるPE P_iが、あるタスク T_kを実行中に、P_i上で将来実行される他のタスク T_iが必要とするデータ D_j を、DTUを使用して D_jを定義したPE P_jのDSMから、P_iのローカルメモリへ転送することをプレロード[8]と言う。また、あるPE P_j上がタスク T_jを実行し、他のPE P_i上のタスク T_iが将来必要とするデータ D_j を定義した時、PE P_jがタスク T_jの実行終了直後だけではなく、他のタスク T_kを実行中に、DTUを使用してPE P_jのローカルメモリからPE P_iのDSMにデータ D_jをストアすることを、ポストストアと言う。ポストストアはタスク T_iの実行が開始される時点までにストアを完了するようスケジューリングされる。データプレロード・ポストストアは、プログラム実行とデータ転送のオーバラップ(同時実行)を実現し、全体としてデータ転送オーバヘッドを最小化するために使用される。

本稿で取り扱うスケジューリング問題は、上述のような先行制約をもつ n 個のタスクを、m 台のプロセッサ上で処理する際に、プレロード・ポストストアも考慮して、全体の処理時間を最小にするスケジュール(割り当て、実行順序)を求める問題である。

2.3　スケジューリングアルゴリズム

データ転送時間を考慮するヒューリスティックアルゴリズムとして、本稿ではCP/DT/MISF(Critical Path/Data Transfer/Most Immediate Successors First)法およびDT/CP(Data Transfer/Critical Path)法[1][5]を用いる。CP/DT/MISF法は次のようなプライオリティ決定法に従うリストスケジューリングの一種で、タスクのPEへの割り当て時に、まずレディ(実行可能)タスク集合からCP(Critical Path)長の最も大きいタスクを選び出し、それらのタスクをデータプレロード・ポストストアを考慮して局所的なデータ転送時間が最小になる組合せでPEを割り当てる。CP長もデータ転送時間も等しいタスクとPEの組合せが複数ある場合は、直接後続タスク数の多いものを優先する。DT/CP法は、ある割り当て時点において、データプレロード・ポストストアを考慮してデータ転送時間が局所的に最小になる組合せでレディタスクをアイドルPEに割り当てる。同じデータ転送時間を要するタスクとPEの組合せが複数あった場合には、CP長の長いタスクを優先する。

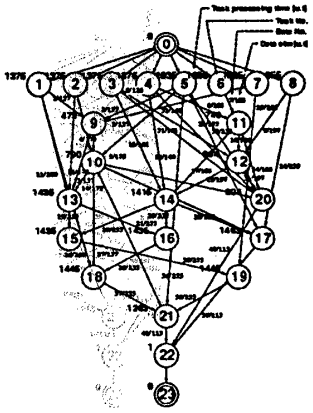

図1: 評価用プログラムのマクロタスクグラフ

このスケジューリングの際、以下のような方法でデータ転送を挿入する。たとえばあるスケジューリング時点 t にタスク T_iのPE P_iへの割当が決定されたとする。マクロタスクグラフのエッジ情報から、タスク T_iが必要とするデータ D_jと、それを定義するタスク T_jが分かる。スケジューリング時点 t

*Data Transfer Overhead Hiding by Data Preloading and Poststoring in Automatic Parallelizing Compiler
†FUJIMOTO Kensaku, KASAHARA Hironori
‡School of Science and Engineering, Waseda University

までに生成されたスケジュールより、T_jの割り当てられたPE P_jがPE P_iと異なる時、タスクT_jの終了時点からスケジューリング時点tまでの間に、PE P_jからPE P_iへDTUによるデータ転送(プレロードまたはポストストア)が可能な期間があれば、この部分にデータD_jの転送を挿入する。挿入できる期間がスケジュール時点tまでの期間にみつからなければ、タスクT_iの実行開始を遅らせて、データ転送を挿入する。あるスケジューリング時点で、ある二つのプロセッサ間でデータ転送が可能かどうかの決定は、PE間の相互結合網のモデルに依存する。

3 通信と処理のオーバラップを考慮した並列化プリプロセッサの実装と性能評価

本節ではプレロード・ポストストアを考慮したスケジューリングアルゴリズムを利用したFORTRANリストラクチャリングプリプロセッサを実装したマルチプロセッサシステムとその上で行なった性能評価について述べる。

3.1 実装対象マルチプロセッサシステム

今回対象とした実マルチプロセッサシステムは、富士通の仮想分散共有メモリ型マルチプロセッサスーパーコンピュータVPP500である。VPP500は最大で222PE構成が可能なMIMDマルチプロセッサシステムである。各PEは最大256Mバイトの SRAM メモリ、LIW RISC プロセッサであるスカラユニットおよびベクトルユニットを持ち、最高処理速度は1.6GFLOPSである。PE間はクロスバネットワークで結合され、各PEの持つデータ転送ユニット(DTU)がPE間通信を処理する。DTUは単方向400Mバイト毎秒のPE間データ転送を行なうことができる。各DTUはPEのスカラユニットおよびベクトルユニットとは非同期的に動作する。このため、データ転送と計算処理は並列に実行される。

3.2 VPP500用のプリプロセッサ

本性能評価では、FORTRANプログラムからVPP500のバイナリを直接出力するコンパイラではなく、並列処理用のコンパイラディレクティブを持つVPP Fortranソースプログラムを出力するプリプロセッサを作成した。ただし以下の評価結果は、マクロタスクグラフより求めたスケジュール結果を効率良く実行するために、VPP Fortranの言語仕様外の動作に依存した同期指定記述を使用した。すなわちポストストア時に、受信側PEがデータ転送終了を待てるよう、非同期データ転送時に転送終了フラグを同時に転送する方式をとっている。

3.3 VPP500上での評価

実装した自動並列化コンパイラでサンプルプログラムを並列化し、実行時間の計測を行なった。評価に使用したプログラムは、DOALL処理可能な複数のループからなる。図1に、そのタスクグラフを示す。マクロタスクグラフのタスク実行時間の総和に対するエッジの重みの総和は、17%程度となる。今回作成した自動並列化プリプロセッサを用い、このプログラムを4PE構成のVPP500上で実行して実行時間計測を行なった。

図2に実行時間の計測結果を示す。実行時間には、プログラム実行の最初と最後に一度ずつ行なわれるforkとjoin動作に要するオーバヘッドが含まれている。図2において、"no overlap"はポストストアによるデータ転送とタスク実行のオーバラップを行なわず、CP/DT/MISF法を用いてスケジューリングした結果である。"CP/DT/MISF with PS"と"DT/CP with PS"はCP/DT/MISF法・DT/CP法にポストストアを加えてデータ転送とタスク実行のオーバラップを行ない、てスケジューリング・実行した結果である。ポストストアを用いた場合は、今回の例題はデータ転送時間がタスク実行時間に比べかなり小さいにもかかわらず、用いない場合と比べ2PE構成で4.1%, 4PE構成で1.6%程度早くなっている。特に2PEの場合のスケジューリング結果では、ほとんど全てのデータ転送がタスク実行中に行なわれ、データ転送オーバヘッドが適切に隠蔽された。

4 むすび

本稿では、VPP500上にデータポストストアを用いたデータ転送オーバヘッドの隠蔽を行なう自動並列化プリプロセッサを実装し、データ転送とタスク実行のオーバラップによるデータ転送オーバヘッド最小化手法の有効性を評価した。今回掲載したデータは、まだプリプロセッサに十分なチューニングを行なっていない状態での値であるため、今後のチューニングによりさらに性能があがるものと期待される。

今後の課題としては、タスクの実行時間とデータ転送時間の予測精度を上げるとともに、データ自動分割手法の開発、条件分岐のあるマクロタスクグラフへの対処法の開発が挙げられる。また、Cenju-3など、異なる実マルチプロセッサシステムへの実装を行なっていく予定である。

本研究の一部は、文部省科学研究費(一般研究(b)05452354, 一般研究(c)05680284)により行なわれた。

最後に、VPP500を使用させて下さった富士通株式会社に感謝致します。

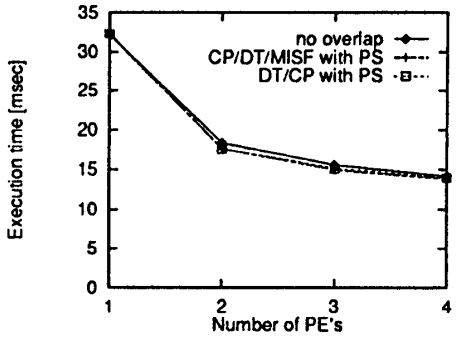

図2: VPP500上での実行時間

参考文献

[1] 笠原博徳: "並列処理技術", コロナ社(1991)

[2] Kasahara H., Honda H. and Narita S.: "A Multi-grain Parallelizing Compilation Scheme for OSCAR", Proc. 4th Workshop on Lang. and Compilers for Parallel Computing (Aug. 1991).

[3] J. J. Hwang, Y. C. Chow, F. D. Anger, and C. Y. Lee, "Scheduling precedence graphs in systems with interprocessor communication times", SIAM J. Comput., pp. 244–257(1989).

[4] Tao Yang, "DSC: Scheduling Parallel Tasks on an Unbounded Number of Processors", IEEE Trans. Parallel Distrib. Syst., vol. 5, pp. 951–967(Sep. 1994)

[5] Kasahara H., Honda H. and Narita S.: "Parallel Processing of Near Fine Grain Tasks Using Static Scheduling on OSCAR", Proc. IEEE Supercomputing '90(Nov. 1990).

[6] Kasahara H. and Narita S.: "Practical Multiprocessor Scheduling Algorithm for Efficient Parallel Processing", IEEE Trans. Comput., C-33, 11, pp. 1023–1029(Nov. 1983)

[7] Coffman E. G.: "Computer and Job-shop scheduling Theory", John Wiley & Sons (1976).

[8] 藤原 和典, 白鳥 健介, 鈴木 真, 笠原 博徳: "データプレロードおよびポストストアを考慮したマルチプロセッサスケジューリングアルゴリズム", 電子情報通信学会論文誌(D-I), J75-D-I, 8 pp.495–503 (1992-8).

[9] 林田 宏一, 藤原 和典, 笠原 博徳: "ローカルメモリを有するマルチプロセッサシステムにおけるデータプレロード・ポストストアアルゴリズム", 電子情報通信学会, 春季全国大会, D-152(1993-3).

[10] 吉田 明正, 前田 誠司, 尾形 航, 笠原 博徳: "Fortran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, pp.1848-1860 (1994-09).