

# マクロデータフロー処理における処理とデータ転送のオーバーラップ

5 F - 6

橋本 茂 藤本 謙作 岡本 雅巳 笠原 博徳  
早稲田大学理工学部電気電子情報工学科

## 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、ループやサブルーチン等の粗粒度タスク間の並列性を利用する粗粒度並列処理（マクロデータフロー処理[2]）、ループのイタレーション間の並列性を利用する中粒度並列処理（ループ並列化）[6]、ステートメントレベルの近細粒度並列処理[5]、さらにそれらを階層的に組み合わせたマルチグレイン並列処理手法[3]が提案されている。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のデータ授受を行なう方法（データローカライゼーション手法[4]）が提案されている。しかし、このようにプロセッサ間データ転送を最小化しても、必要なデータ転送が残る。また、多くの計算機ではデータ転送をCPUの代わりに行なうデータ転送ユニットを各プロセッサエレメントに持たせ、CPUの処理と平行してデータ転送を行なえるようになっていて、見かけ上のデータ転送オーバーヘッドを隠蔽することが可能である。しかしながら、ユーザが処理とデータ転送のオーバーラップを考慮したプログラミングを行なうことは困難である。

本稿では、マクロデータフロー処理において、処理とデータ転送を効率的にオーバーラップさせデータ転送オーバーヘッドを削減するための、マクロデータフロー処理におけるデータプレロード[1]手法を提案する。本手法を適用することにより、対象マクロタスクの割り当てを待たずにデータロードが行なえるようになり、処理とデータ転送を効率的にオーバーラップさせることが可能となる。

## 2.マクロデータフロー処理

マクロデータフローコンパイルーション手法[2][6]では、FORTRANプログラムをBPA（疑似代入文ブロック）、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。

コンパイラは次に、BPA、RB、SB等のMT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MT間並列性を最大限に引き出すために、各MTの最早実行可能条件解析[6]を行なう。この各MTの最早実行可能条件は、マクロタスクグラフと呼ぶ無サイクル有向グラフで表わすことができる。

最後に、コンパイラはMTを実行時にプロセッサに割りあ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。実行時のダイナミックスケジューリングアルゴリズムとしては、Dynamic-CP法[2]が提案されている。Dynamic-CP法では、コンパイル時に各MTの推定処理時間と条件分岐を含む場合は分岐確率により各MTからマクロタスクグラフの出口ノードまでの最長パス長(CP長)を計算することにより各MTに割り当て優先順位を付け、実行時に複数のMTが実行可能な場合にはその優先順位が高いMTからプロセッサエレメント(PE)に割り当てる。

## 3.MTの先行割り当てによる処理とデータ転送のオーバーラップ

本章ではマクロデータフロー処理において、MTの先行割り当て手法を用いて処理とデータ転送のオーバーラップを実現するための手法について述べる。

### 3.1.レディMTの先行割り当て

マクロデータフロー処理のダイナミックスケジューリングオーバーヘッドを軽減するための手法として、MTの先行割り当てが提案されている。この手法は、最早実行可能条件の満たされたMT(レディMT)の割り当てを、その時点でのアイドルPEに対してのみ行なうのではなく、全てのPEを対象に行なうというものである。その際、割り当てられているMTの推定実行時間の和の少ないPEに優先してレディMTを割り当てる。また、後から割り当て優先順位の高いものがレディになり実行時間が延びるのを避けるため各PEに先行割り当てされるMTの数の制限も行われる。

### 3.2.ロード・ストア命令の非同期実行

CPUの処理とDTUのデータ転送をオーバーラップさせる場合は、MTからロード・ストア部分を独立させる必要がある。そこで本手法では図1に示すように、ロード命令発行部、ストア終了確認部をMTから切り離し、ロード終了確認部と処理部及びストア命令発行部をMTに残している。

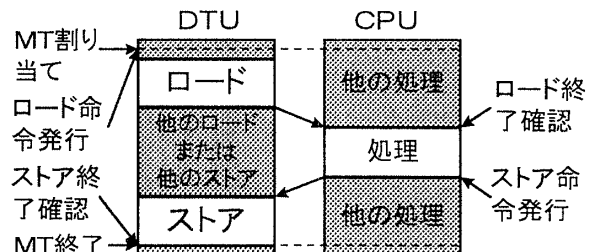


図1 処理とデータ転送のオーバーラップ

ロード命令は対象MTの先行割り当て時に発行し、対象MTの処理はCPUがアイドルになりDTUのロード終了が確認できた時点で開始する。ストア命令は対象MTの処理終了時に発行し、CPUはDTUのストア

Overlapping of Macro Task Processing and Data Transfer for Macro-dataflow Computation, HASHIMOTO Shigeru, FUJIMOTO Kensaku, OKAMOTO Masami, KASAHARA Hironori, Dept. of Electrical, Electronics and Computer Engineering, Waseda Univ.

の終了を待たずに MT 処理を終了する。従って MT の CPU 占有時間は、MT の処理開始から処理終了までとなる。ただし、スケジューラにおける対象の MT 終了は対象 MT の全てのストアが終了した時点とする。

### 3.3 処理とデータ転送のオーバーラップ

以上の変更によりロード・ストア時に MT が CPU を占有することがなくなり、処理とデータ転送をオーバーラップさせることが可能になる。対象 MT の先行割り当てが行われ、DTU で対象 MT へのデータロードを実行する際に、CPU で他の MT の処理を実行すれば、ロードと処理はオーバーラップしていることになり、対象 MT の処理終了して、DTU で対象 MT からのデータストアを実行する際に、CPU で他の MT の処理を実行していれば、ストアと処理はオーバーラップしていることになる (図 1)。

## 4. マクロデータフロー処理におけるデータプレロード

MT の先行割り当てによる処理とデータ転送のオーバーラップ手法では、ロードすべきデータがストアされてから対象 MT が先行割り当てされるまでの間はデータをロードすることができず、対象 MT が先行割り当てされなければオーバーラッピングスケジューリングを行えない問題があった。

これに対し本稿で提案するデータプレロード手法では、これらの問題点を解決するために前章の手法に加え、ロード命令に実行開始条件を持たせることにより、MT の最早実行可能条件成立前にデータロードを実行することを可能にし、効率的な処理とデータ転送のオーバーラップを実現する。

### 4.1. ロードの実行開始条件

データロードをロードすべきデータが共有メモリにストアされてから対象 MT が実行されるまでの間に行なうために、ロード命令に実行開始条件を持たせる。データロードは、データが共有メモリ上にありロードするデータが使用されることが確定しないと行なうことができない。したがって実行開始条件は、対象 MT の実行が確定し関連するデータのストアが終了することである。

### 4.2. MT の実行 PE 決定について

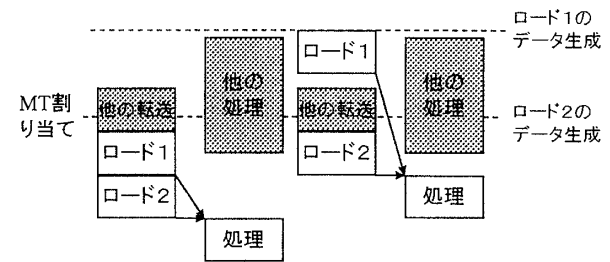
ロード命令の発行が対象 MT の割り当て前になる場合は、対象 MT の実行先 PE を決定する。ただし、実際の割り当ては MT の最早実行可能条件が成立したときに行なう。対象 MT の実行先は、割り当てられた MT の実行時間推定値の和と実行先の確定した MT の実行時間推定値の和が最も少なく、必要なデータをローカルメモリ内に格納できる PE とする。

また、MT の実行先 PE 決定時刻と実行開始時刻の間の差が大きいとき、あるいは多くの MT の実行先 PE が最早実行可能条件の成立を待たずに決定している場合には、負荷分散の点で問題が生じることがある。そのため、先行割り当てされる前に実行先 PE が決定している MT の数を制限する必要がある。

### 4.3. データプレロードの様子

データプレロードの様子を図 2 に示す。データプレロードを適用しない場合は、MT が割り当てられるまでデータロードを行えないためロード 1 が実行可能となってもロードすることができない。それに対し、データプレロードを適用した場合には、ロード 1 のデータが生成された時点でロード 1 が実行可能となるた

め、DTU の空き時間を効率的に使用することができるようになり、ロードと処理がオーバーラップする機会も増える。



プレロードを適用しない場合 プレロードを適用した場合

図 2 データプレロードの様子

## 5. 性能評価

本手法のシミュレーションによる評価を行なった。今回の評価では PE が 3 台で各 PE がローカルメモリ及び DTU を持ち、集中共有メモリを持ったアーキテクチャを対象とする。ただし、DTU は送受信を同時には行なえないものとする。マクロタスクグラフは、MT 数 50 個、データ転送比 50%、分岐数 2 個の条件でランダムに生成したもの 100 個を使用した。また、1PE 当たりの先行割り当てされる MT の最大数を 5 個、割り当て前に実行先 PE の決定している MT の最大数を 5 個とした。その結果、データプレロードを適用したものは、処理とデータ転送のオーバーラップを適用しないものと比較して平均で 7.4%、処理とデータ転送のオーバーラップのみを適用したものに比べ平均で 3.2% 実行速度が向上することが確認された。

## 6. まとめ

本稿では、マクロデータフロー処理におけるデータプレロード手法について述べた。提案手法では、ロード命令に実行開始条件を持たせることにより、処理とデータ転送を効率的にオーバーラップさせることができる。今後、現在開発中のマルチターゲットマシンコンパイラにインプリメントを行ない実機上で評価を行なう予定である。

本研究の一部は、文部省科学研究補助金 (一般研究 (C) No.07680372) により行われた。

## 参考文献

- [1] 藤原,白鳥,鈴木,笠原:“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ス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 (1988).
- [2] 本多,合田,岡本,笠原:“Fortran プログラム粗粒度タスクの OSCAR における並列実行方式”,信学論 D,Vol.38,No1 (Jan. 1989).
- [3] 岡本,合田,宮沢,本多,笠原:“OSCAR マルチグレインコンパイラにおける階層型マクロデータフロー処理”,情処論,Vol.35(Apr. 1994).
- [4] 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 (Sep. 1994).
- [5] 笠原:“マルチプロセスシステム上での近細粒度並列処理”,情報処理 Vol. 37, No.7,pp.651-661(Jul. 1996).
- [6] 笠原:“並列処理技術”,コロナ社.