4J-06 メモリ容量を考慮したデータプレロード マルチプロセッサスケジューリング

増田 高史, 飛田 高雄, 舟山 洋央, 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

分散メモリあるいは分散共有メモリを持つマルチプロセッサシステムにおいて，効率の良い並列処理を行うためには，分散 (共有) メモリへのデータの自動分割配置を伴うプロセッサ間データ転送オーバーヘッドの最小化が必須である [1]．このため，ユーザ指定によるデータ分割・配置のための High Performance Fortran [2]，ループ内の作業配列をローカル化する Array Privatization 法 [3]，データローカライゼーション手法 [4] 等の研究が活発に行われている．しかし，最良のデータ配置が決定しても全データ転送を除去することは不可能であり，除去出来なかったデータ転送オーバーヘッドを，データプレロード・ポストストア手法により，CPU 上でのタスク処理とオーバーラップさせることで隠蔽することが重要となっている [5]．処理とデータ転送のオーバーラップに関しては，データキャッシュにおけるデータプリフェッチ [6] に関する研究も活発である．これはプリフェッチ命令の挿入等によるキャッシュラインレベルにおけるデータ転送を想定している．しかしデータプレロードに関する従来の研究では，実際にデータの置かれるローカルメモリの容量制限に関しては考慮されておらず，メモリの容量に制限のある場合には適用が難しいという問題があった．

本稿では，スタティックスケジューリングを対象とし，ローカルメモリの容量に制限がある場合でもデータプレロード手法を有効に活用できるスケジューリングアルゴリズムを提案し，シミュレーションによる性能評価について述べる．

2 メモリ容量を考慮したスケジューリングシミュレーションモデル

本節では，本稿で対象とするマルチプロセッサシステムのモデルとタスクグラフの表現，ローカルメモリの容量制限の方法，データプレロードを適用したデータ転送の割り当て手法に関して述べる．

2.1 対象マルチプロセッサシステムモデル

ここでは対象マルチプロセッサのモデルとして，相互結合網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) とデータ転送ユニット (DTU) を持ち，さらに各 PE からアクセス可能な集中共有メモリ (CSM) を持つようなマルチプロセッサシステムを考える．DTU によるデータ転送は，CPU 上でのタスク処理とオーバーラップして行うことが可能で，さらにリモートリード・ライト双方向のデータ通信を同時に行うことが可能であるとする．

2.2 対象マクロタスクグラフモデル

図 1に本稿で対象とするマクロタスクグラフ [1] を示す．図中各ノードはタスクを表し，ノード内の数字はタスク番号，ノードの左側の数字はタスクの実行時間を表している．ノード間のエッジはデータ依存を表しており，各エッジにはデータ転送時間，データ番号，データサイズが付加されている．

2.3 ローカルメモリの容量制限

ここでは，各タスクで使用されるデータは，CPU 上でのタスク処理前に DTU によって LM にロードされ，そのデータの量は，マクロタスク上に示されるデータサイズに基づいて求めることが可能であるとする．すなわち，タスク及びデータ転送の割り当てを行う場合には，対象となる全てのスケジューリング時点における LM 容量のチェックを行い，LM 容量を超えない場合に限り割り当てが可能となるものとする．

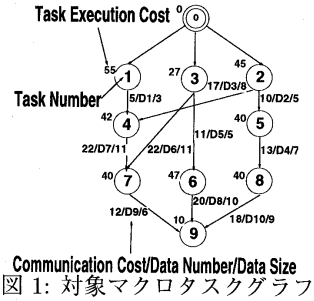

図 1: 対象マクロタスクグラフ

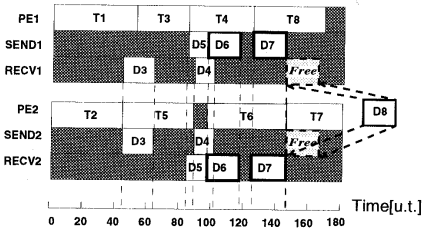

図 2: スケジューリングイメージ

図 2は横軸に時間をとり，各 PE の処理状況，DTU の send unit，receive unit の動作状況，すなわち生成されるスケジュール結果を表している．図 2において，PE_2 から PE_1 へのデータ転送 D_8 を図中 *Free* に割り当てる場合を考える．この時，挿入されるべき転送 D_8 によって新たにメモリを必要とするのは PE_1 であるから，*Free* の開始時刻 146[u.t.](unit time) における PE_1 の LM データ量を求め，容量が不足しないかどうかのチェックを行う．

また，スケジューリング過程において LM 容量が足りない場合，一時的に不要になっているデータを LM から CSM に退避させることも可能であるとする．但しその場合には，LM から CSM へのストアと，後にデータが参照される時に CSM から LM への再ロードが必要になり，オーバーヘッドになる．

2.4 容量制限がある場合のデータプレロード

従来の容量制限を考慮しない場合，すなわち大容量の LM があると仮定した場合のデータ転送は，DTU 及びネットワークの空きだけを見て割り当てられていたが，本研究のように容量制限がある場合には，LM 空き容量のチェックを行う必要がある．従って容量制限下でのデータプレロードとは，あるスケジューリング時点において，タスク T_i をプロセッサ PE_k に割り当てる場合，そのデータ依存先行タスク T_j の実行終了時刻から T_i の実行開始時刻までの間で，T_j を実行したプロセッサ PE_m が T_j が生成したデータ D_j を PE_k の LM 容量の制限値を超えないタイミングで，他のタスク T_l の実行とオーバーラップさせて PE_k の LM へロードすることである．

図 2において，時刻 146[u.t.] に PE_2 に割り当てられているタスク 7(T_7) に注目する．T_7 の先行タスクは T_3 と T_4 であ

---

* A Multiprocessor Scheduling Scheme with Data Preload Considering Memory Capacity
Takafumi Masuda,Takao Tobita,
Hirochika Funayama,Hironori Kasahara
Department of Electrical, Electronics and Computer Engineering, Waseda University

り，T_3 から T_7 へのデータ転送は時刻 93[u.t.] から 115[u.t.] の D_6，T_4 から T_7 へのデータ転送は時刻 124[u.t.] から 146[u.t.] の D_7 によって行われており，ここでは D_6 のデータ転送はタスクの処理 T_4 と，D_7 は T_8 とオーバーラップして行われていることが分かる．

3 容量制限を考慮したスケジューリングアルゴリズム

本節では，データプレロード手法を伴うローカルメモリの容量を考慮したスケジューリングアルゴリズム，ETF/CP with Data Preload Considering Memory Capacity (ETF/CP wPL_MC) 法を提案する．ETF/CP wPL_MC 法は以下の手順で構成される．

1. 各タスクから出口ノードまでの最長パス (CP) 長を求める．
2. CP 長の大きい順に，プライオリティリストを作成する．
3. 割り当てが未終了のタスク集合から，全ての先行タスクが割り当てられているもの (レディタスク集合) を選択する．
4. 各レディタスクを各 PE に割り当てた場合の最早実行終了時刻を，データプレロード，メモリ容量を考慮して計算する．この時プレロードを考慮したデータ転送の挿入は次のように行われる．
   - 4-1. 対象データ依存先行タスクの終了時刻から割り当て時刻までの間で，DTU 及びネットワークがデータ転送所要時間以上連続で空いている転送可能候補を求める．
   - 4-2. 各転送可能候補について時刻の早いものから順にメモリ容量のチェックを行い，割り当て可能であればその転送可能候補を割り当て時刻とする．全ての転送可能候補で容量制限を超える場合は手順 4-3 を行う．
   - 4-3. 生死解析の結果に基づいて，生きているデータの内一時的に不要なデータを求め，対象データ転送が必要とする最低限のメモリ容量を確保するために，CSM への一時的なデータ退避を行う．
5. 最早実行終了時刻の最も早いタスクと PE の組合せを一個だけ選択し，その組み合わせでタスクを PE に割り当てる (ETF)．候補が複数ある場合は，2 で作成したプライオリティリストでの優先順位の高いものを割り当てる．
6. 3 から 5 を割り当て対象タスクがなくなるまで繰り返す．

4 性能評価

本章では，ランダムマクロタスクグラフを用いたシミュレーションによる，提案アルゴリズムの性能評価について述べる．

今回のシミュレーションでは，スケジューリングアルゴリズムの公平な評価のためのタスクグラフセットである Standard Task Graph Set[7] に，各タスクで定義・参照されるデータ情報を追加した 1000 例のランダムタスクグラフを用意した．

本稿で提案する ETF/CP wPL_MC 法と，単純な FIFO アルゴリズム，データプレロードを考慮したヒューリスティックアルゴリズム CP/ETF 法，さらに本手法においてデータプレロードを適用しないアルゴリズムに関して，PE 数を 2，4，8 台と変化させた場合の，PE 数 1 台での逐次実行時間に対する平均速度向上率を図 3に示す．ここでローカルメモリ容量は，各タスクグラフ毎にそのタスクグラフ中で使用される最大のデータ量と設定している．

図3において PE 数 8 台の場合で，本手法 (プレロード有り)，FIFO，CP/ETF の平均速度向上率はそれぞれ 5.31 倍，3.72 倍，4.70 倍であり，本手法は FIFO に対し約 43%，CP/ETF 法に対し約 13% 優れた性能を示している．また，本手法にデータプレロードを適用しない場合の平均速度向上率は，PE 数 8 台の場合で 4.62 倍であり，本手法にデータプレロードを適用した場合，適用しない場合より約 14% の速度向上が得られ，本手法の有効性が確認された．

5 まとめ

本稿では，タスクの処理とデータ転送をオーバーラップさせることによりデータ転送オーバーヘッドを隠蔽するデータプレロードを，メモリ容量を考慮して実現するマルチプロセッサスケジューリングアルゴリズム ETF/CP wPL_MC 法を提案した．シミュレーションによる性能評価の結果，本手法により他のアルゴリズムより優れたスケジューリングが得られ，メモリ容量に制限がある場合でもデータプレロードを有効に利用できることを示した．

今後の課題としては，提案したアルゴリズムの実マシン上での評価等が挙げられる．

本研究の一部は日本学術振興会未来開拓学術研究推進事業知能情報・高度情報処理第5プロジェクト分散・並列スーパーコンピューティングのソフトウェアの研究 (JSPS-RFTF96P00505) 及び通産省次世代情報処理基盤技術開発事業並列分散分野マルチプロセッサコンピューティング領域研究の一環として行われた．

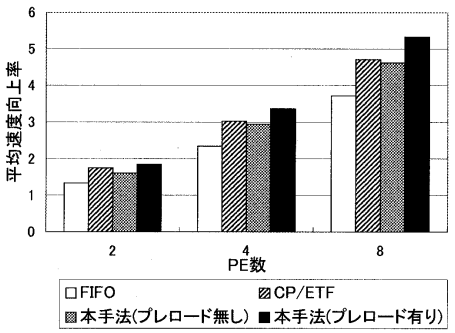

図 3: 各アルゴリズムの平均速度向上率

参考文献

[1] 笠原博徳. 並列処理技術. コロナ社, 1991.

[2] HPF Forum. *High Performance Fortran Language Specification.*

[3] P. Tu and D. Padua. Automatic array privatization. In *6th Annual Workshop on Languages and Compiler for Parallel Computing*, 1993.

[4] H. Kasahara and A. Yoshida. A data-localization compilation scheme using partial static task assignment for fortran coarse grain parallel processing. *Journal of Parallel and Distributed Computing*, May 1998.

[5] 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. 信学論, Vol. J75-D-I, No. 8, pp. 495–503, August 1992.

[6] T.C.Mowry and M.S.Lam and A.Gupta. Design and evaluation of a compiler algorithm for prefetching. *Proc.Fifth Int'l Conf.Architectural Support for Programming Languages and Operating Systems*, pp. 62–73, April 1992.

[7] T. Tobita and H. Kasahara. A standard task graph set for fair evaluation of multiprocessor scheduling algorithms. In *Proc. ICS99 Workshop*, pp. 71–77, June 1999.