

先行評価を用いたマクロタスクの多段仮実行方式における分散制御手法

2L-6

山名早人 安江俊明 石井吉彦 村岡洋一
早稲田大学理工学部

1. まえがき

本報告では、先行評価を用いたマクロタスクの多段仮実行方式[1]におけるマクロタスクの効果的な制御手法として、マクロタスクの分散制御手法を提案する。

多段仮実行方式は、プログラム中のデータ依存と制御依存の内、データ依存を保証した段階でマクロタスクと呼ぶタスクの実行を開始し、後で制御依存に基づいて制御確定したマクロタスクを選択する手法である。本方式を実際のマルチプロセッサ上で実現するにあたっての問題点は、実行時に発生する各種オーバーヘッドの削減である。実行時のオーバーヘッドには、制御が確定しない段階で実行を開始することにより発生する(1)メモリバンド幅の増大に起因するオーバーヘッド、(2)多数のマクロタスクを制御するために発生する制御オーバーヘッドがある。本稿では、これら2つのオーバーヘッドの内、(2)のオーバーヘッドを削減するための手法として、プロセッサにマクロタスク制御専用のハードウェアを付加し、集中制御を廃したマクロタスクの制御手法を提案する。(1)の問題は、マクロタスクのスケジューリング問題であり、今後の課題である。

2. 多段仮実行方式[1]

多段仮実行方式とは、プログラム中のデータ依存と制御依存の内、データ依存を保証した段階でマクロタスクと呼ぶタスクの実行を開始し、後で制御依存に基づいて制御確定したマクロタスクを選択する手法である。制御依存を越えた実行方式としては、従来、Boolean Recurrence[3]を持つループを対象に、研究が行われてきた[2][3][4]。[3]では、条件分岐を先行評価することによって、平均5.1倍(条件分岐を無限に先行評価した場合)の性能向上が得られることをいくつかのアプリケーションに対してシミュレーションを行い示している。しかし、[2][3][4]では、対象プログラムが限定されており、かつ、特種なループに対する実行方式のみが提案されている。

これらの問題に対して我々は、多段仮実行方式を提案し、マクロタスク間の先行評価手法として一般化している[1]。マクロタスク生成においては、制御フロー上の直前に位置するマクロタスクからのデータ依存が存在しないように構成することによって、(1)マクロタスクの粒度を上げ、その結果として、(2)先行評価の効果の小さい部分を1マクロタスクに融合することができ、かつ、(3)先行評価効果の大きい条件分岐部分をマクロタスクの分割の切れ目とすることができる。図1にマクロタスク生成の例を挙げる。図1の例では、3つのマクロタスクが生成される。ここで、マクロタスク生成手順については、文献[1]を参照されたい。図1に示すように、各マクロタスク(MT)の制御の入点(A,B,C)においては、制御フロー上の直前のマクロタスクからのデータ依存が存在しない。

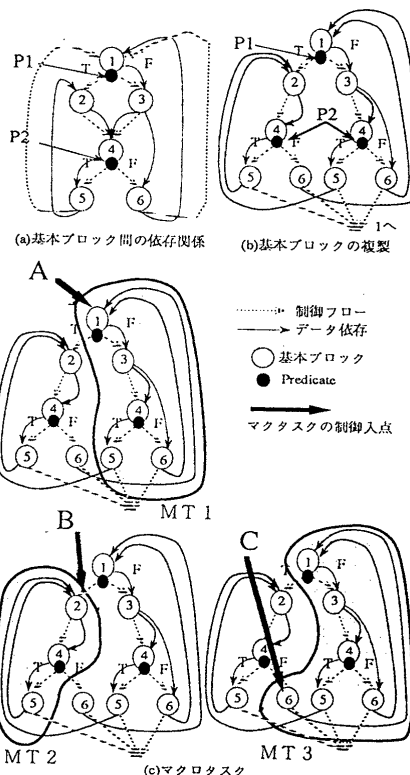


図1 マクロタスク例

3. マクロタスクの制御

図1で示されるようなマクロタスクを先行評価するためには、以下の3制御を行わなくてはならない。

- (1) マクロタスクへの入力データの定義保証後の起動
制御が未決定時点での起動であり、仮実行と呼ぶ。また、仮実行中は共有メモリへのデータ出力を行わない。起動条件は、マクロタスクへの入力データの定義保証のみである。
- (2) 制御確定後の仮実行から本実行への移行
本実行移行後、共有メモリへのデータ出力を行う。
- (3) 制御が確定しなかった場合の仮実行停止

これら(1)-(3)の制御を集中制御で行った場合、オーバーヘッドが顕著に表れてしまう。そこで、(1)(3)の制御を各プロセッサに分散させる手法を以下で提案する。

3.1 マクロタスクの起動

マクロタスク(MT)を起動するためには、そのMTが必要とする全てのデータの定義保証をしなくてはならない。ここで、あるMT-Aが、MT-Aに先行するMT-Bからのみデータ依存を受けている場合には、MT-Bにおけるデータ定義終了のみをトリガとして後続のMT-Aを起動することができる。この場合、「データの定義終了→新たなMTの起動」という手順をMT内で処理でき、集中したタスク管理機構が不用である。以下では、このように特定のMTのみからデータ依存を持つMTと、複数MTからデータ依存を持つMTとを分類する。そして、マクロタスクの起動をプロセッサに分散させるために、マクロタスク起動コードをマクロタスク内に埋めこむ手法を示す。

- ・データ依存を持たないMT 他どのMTからもデータ依存を持たない。このため、いつでも起動できる。この場合、最大先行評価段数Nを定義し、次起動がN内であれば起動させる。また、起動は、ダミーのMTを1つ作成し、ダミーMTから起動させる。
- ・1つのMTからのみデータ依存を持つMT 先行するある1つのMTからのみデータ依存を持つ。このため、ソースとなるMTから起動ができる。この場合、データ依存のソースとなるMT内にマクロタスク起動コードを埋めこむ。
- ・複数のMTからデータ依存をへ持つMT 先行する複数のMTからデータ依存を持つ。このため、複数のMTの終了を待つ必要がある。複数のMTにおけるデータ定義終了を判定するために、データ駆動を用いる。図2に示すようなデータ待合せ機構を用意し、データの待合せを行う。マッチングのためのタグとして、3.2節で述べる経路情報を用いる。

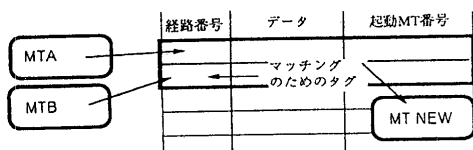


図2 データ待合せ機構

図3に、図1のMTに上記の処理を行った場合の結果を示す。

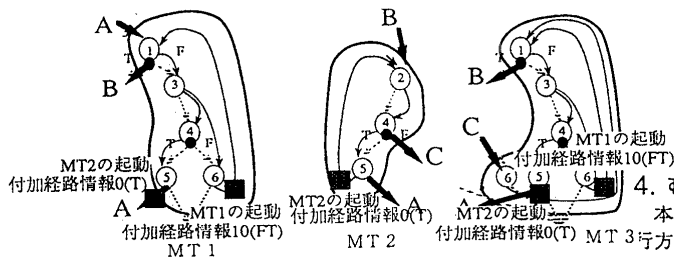


図3 MT起動コードが埋めこまれたMT

3.2 マクロタスクの制御

3.1で作成されたMTを制御するために、以下の2つのグローバル情報と、最大先行評価段数Nを用いる。
レベル番号 制御が通過した条件分岐数。初期値はレベル1。
経路情報 現在制御が確定しているMT(ルートMTと呼ぶ)から自MT(仮実行中)までの間に存在する条件分岐の結果を1の次に0(true), 1(false)を羅列したもの。例えばT/F/Tであれば1010。
 各MTは、これらの情報を用いてMTの制御を行う。以下に、各MT内での処理を示す。なお、以下の処理は、オーバーヘッド軽減のため、ハードウェアで用意するものとする。
ルートMTにおける処理:

自MT内において条件分岐が処理される毎にレベル番号をインクリメントすると同時に、(レベル番号, 評価結果)を放送する。こ

で、評価結果とは、ture/falseをそれぞれ0/1で示したものである。全てのMT及びデータ待合せ機構における処理:

以下、待合せ機構における処理では、自MTを待合せ機構内のデータと読み替えるものとする。

- (1) ルートMTから放送される評価結果(0/1)と、自MTが持つ経路情報の第2bit目を比較し、一致すれば第2bit目を削除し、一致しなければ自MTが選択されなかったことを示すので、仮実行を停止する。例えば、自MTの経路情報が1011であり、ルートMTから0が放送された場合には、経路情報を111に更新する。また、ルートレベル番号も更新する。
- (2) MT起動コードが存在する場合には、(3)以下の処理を行う。そうでなければ、(1)を繰り返す。最終的に、自MTの経路情報が1になった場合には、制御が自MTに移ったことを意味するので、仮実行から本実行に移行し、ルートMTの処理を行う。
- (3) 新MT起動に必要なデータの定義終了時点で、新MTを起動する。この際、(ルートレベル番号, 新経路情報)を付加情報として新MTに付加する。ここで、新経路情報とは、自MTの経路情報に、自MTから新MTまでの経路情報を付加したものである。例えば、自MTの経路情報が1010で、自MTから新MTまでの経路情報が11の場合には、101011となる。この際、(経路情報のビット長-1)が新MTの先行評価段数に一致するので、この値が最大先行評価段数Nより大きい時には、起動をさせない。

起動直後のMTにおける処理:

(ルートレベル番号, 新経路情報)の情報を持ちMTが起動されると、まず、経路情報を修正する。プロセッサが持つルートレベル番号(各プロセッサは、ルートMTから放送されたレベル番号と評価結果の履歴を持つとする)とMTの持つルートレベル番号が一致するまで、上記(1)の処理を行う。これは、新MTが起動中に発生したルートMTからの制御を反映させるためである。

以上述べた制御手法を用いることによって、プロセッサに分散したマクロタスク制御を行うことが可能となる。図4に、図1のMTを例とした場合の制御概要を示す。

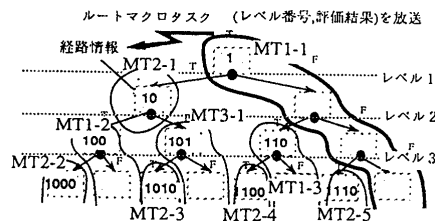


図4 分散制御概要図

4. むすび

本報告では、先行評価を用いたマクロタスクの多段仮実行方式におけるマクロタスクの効果的な制御手法として、集中制御を廃したマクロタスクの制御手法を提案した。本方式は、実行時の制御オーバーヘッドを削減するのが目的である。実行時のオーバーヘッドとして、他に、メモリバンド幅の増大に起因するオーバーヘッドが存在するが、これについては、今後の課題である。

参考文献

- [1] 山名, 安江, 石井, 村岡: "先行評価を用いたマクロタスクの多段仮実行方式の提案", JSSP92予稿集, pp.117-122, 1992
- [2] E.Riseman et al.: "The Inhibition of Potential Parallelism by Conditional Jumps", IEEE Trans. on Comp., pp.1405-1411, 1972
- [3] U.Banerjee et al.: "Fast Execution of Loop with IF statements", IEEE Trans. on Comp., pp.1030-1033, 1984
- [4] A.K.Uht: "Requirements for Optimal execution of Loops with Tests", Proc. of ICS'88, pp.230-237, 1988