4H−9

ＯＳＣＡＲ上での ＦＯＲＴＲＡＮプログラムの 階層的マクロデータフロー処理手法

小椋　章央、合田　憲人、本多　弘樹、笠原　博徳、成田　誠之助
早稲田大学　理工学部　電気工学科

１．はじめに

マルチプロセッサ上でFORTRANプログラムの並列処理をする場合には、各プロセッサへの割当て単位であるタスクの大きさをどの様に選ぶかによって、プログラムに内在する並列性をどれだけ有効に引き出すことができるかが決ってくる。現在までに筆者等は、ステートメント等をタスクとする細粒度タスクのスタティックスケジューリングを用いた並列処理、Ｄｏａｌｌ等のＤｏループイタレーションをタスクとする中粒度並列処理、ＤＯループあるいは基本ブロックなどをタスクとする粗粒度タスク（マクロタスク）のダイナミックスケジューリングによる並列処理（マクロデータフロー処理）とを組み合わせた階層的な並列処理手法を提案し、マルチプロセッサシステムＯＳＣＡＲ上でその有効性を検証してきた[1]-[6]。

本稿では、粗粒度レベルにおけるマクロデータフロー処理を各マクロタスクの並列性により階層的に適用する方式について述べる。

２．ＯＳＣＡＲのアーキテクチャ

図１にＯＳＣＡＲのアーキテクチャを示す[1]。ＯＳＣＡＲは最大１６台のＲＩＳＣプロセッサ（ＰＥ）と、ダイナミックスケジューラ及びホストコンピュータとのインターフェースして利用されるコントロール＆ＩＯプロセッサ（ＣＩＯＰ）とが、３モジュールのコモンメモリ、各ＰＥ上の分散された共有メモリと３本のバスによって結合されている集中及び分散共有メモリ型のマルチプロセッサシステムである。

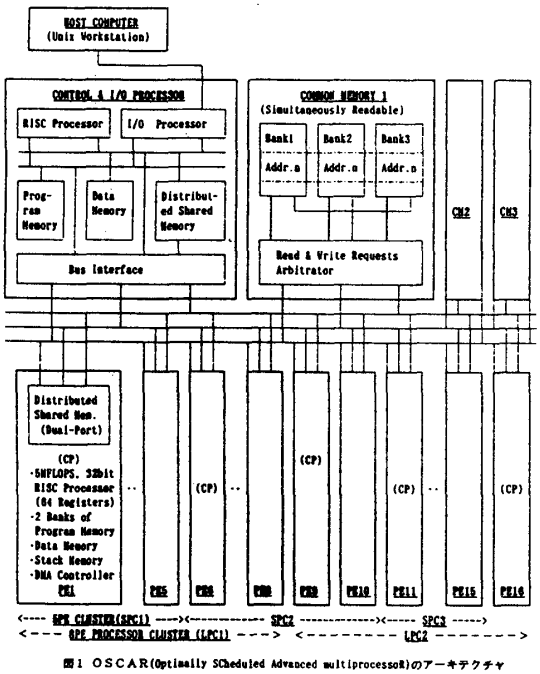


図1 OSCAR(Optimally SCheduled Advanced multiprocessoR)のアーキテクチャ

また、ＯＳＣＡＲは本質的には、平等型のマルチプロセッサシステムであるが、図１のように、８台のＰＥに一つのバスを割当て、２つのプロセッサクラスタシステムをシミュレートしたり、５台のＰＥに一つのバスを割り当てて３つのプロセッサクラスタからなる階層的なマルチプロセッサシステムをシミュレートすることができる。

３．階層的分割されたマクロタスクの並列処理手法

マクロタスクを構成するものとして主に次の３つのブロックがある。

（１）ＢＰＡ (Block of Psuedo Assignment statement)
基本ブロックあるいは複数の基本ブロックを融合したもの。

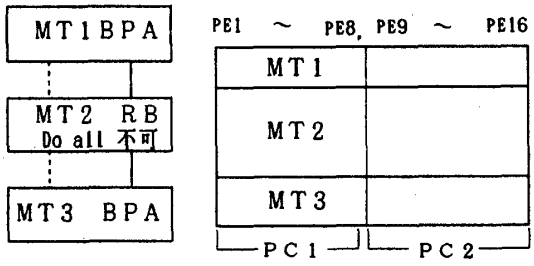
（２）ＲＢ (Repetition Block)
Ｄｏループなどの繰り返しブロック。

（３）ＳＢ (Subroution Block)
インライン展開が効果的に適用できないサブルーチンにより構成されるブロック。

また、上記のブロックを複数融合した、融合的なマクロタスクを定義する事ができる。

まず、単純な単階層のマクロデータフロー処理を用いて、図２（ａ）のマクロフローグラフのように、プログラムがマクロタスク１（ＭＴ１）とＭＴ３のような基本ブロック（ＢＰＡ：代入文からなるブロック）とＭＴ２のようなループ（ＲＢ）からなるブロックを処理する場合には、ＭＴ１からＭＴ２、ＭＴ２からＭＴ３にデータ依存があるため、これらのマクロタスクを２つのプロセッサクラスタを用いてダイナミックにプロセッサクラスタに割り当てて処理をしようとすると、図２（ｂ）のように全てのマクロタスクが一つのプロセッサクラスタのみに割り当てられてしまう。すなわち、単一階層のマクロデータフロー処理では、この例のような場合一つのプロセッサクラスタ上での処理しかできず、全プロセッサを効率よく利用できないという問題がある。このような単階層マクロデータフロー処理の欠点を克服するためには、マクロタスクを階層的に分割しそれらのサブマクロタスクの階層的なデータフロー処理を行う階層的なマクロデータフロー処理手法が考えられる。

この階層的マクロデータフロー処理について、図３のマクロフローグラフを持つプログラムをＯＳＣＡＲ上で処理する場合を例に取り簡単に説明する。



（ａ）マクロフローグラフ（ｂ）プロセッサへのスケジューリング

図２　単階層でのマクロデータフロー処理

A Hierarchical Macro-dataflow Computation of Fortran Program on OSCAR
Akio OGURA,Kento AIDA,Hiroki HONDA,Hironori KASAHARA,Seinosuke NARITA
WASEDA UNIV.

図３の例では、ＭＴ１、ＭＴ２、ＭＴ３に関してはデータ依存があることによりシーケンシャルな実行しかできない。そこで、１６台の全ＰＥを一つのプロセッサクラスタと考え、図３（ａ）のように、スタティックに割当てる。このとき、ＭＴ１、及びＭＴ３は基本ブロックであるので、その処理には、１６台のＰＥを用いたスタティックスケジューリングによる細粒度の並列処理が適用される。また、Ｄｏａｌｌ不可能なＲＢ（Ｄｏループ）であるＭＴ２に対しては、内部の並列性を活かすために図３（ｂ）のように、階層的マクロデータフロー処理を適用する。このとき、ＭＴ２の内側では、ＭＴ２．１～ＭＴ２．６の６つのマクロタスクが生成され、そのマクロフローグラフには、条件分岐が存在するのでこれらのマクロタスクの処理にはＰＥ１～ＰＥ８、ＰＥ９～ＰＥ１６で２つのプロセッサクラスタを構成してダイナミックスケジューリングを用いる。このマクロフローグラフをダイナミックスケジューリングを用いて並列処理する際に、ＭＴ２．１からの条件分岐によりＭＴ２．４側のパスが選ばれた場合は、ＭＴ２．４はＤｏａｌｌ可能であるのでＰＥ１～ＰＥ８で構成されたプロセッサクラスタ１（ＰＣ１）を用いてループイタレーションレベルでの中粒度の並列処理が適用される。また、ＭＴ２．５はＤｏａｌｌ不可能なＲＢであるので、ＭＴ２．５が割り当てられたＰＣ２．２に含まれるＰＥ１２～ＰＥ１６の８台のＰＥでＰＣ２．１，ＰＣ２．２を構成し、さらに内側を階層的な処理を行なう。ＭＴ２．５の内側のマクロフローグラフには、条件分岐が存在せず、各マクロタスク間のデータ依存のみが存在するので、ＭＴ２．５．１からＭＴ２．５．４は、図３（Ｃ）のように、スタティックにプロセッサクラスタに割り当てられ、ＰＣ２．１及び２．２の内部のＰＥを用いたスタティックスケジューリングによる細粒度の並列処理が適用される。

以上のように、階層的マクロデータフロー処理では、各階層でのマクロタスクのクラスタへの割当てをスタティックスケジューリングとダイナミックスケジューリングを効果的に選択し組み合わせて行なうことにより、より高い並列性を抽出し、並列処理効果を上げることが可能となる。

マクロタスクの階層的分割を行う場合には、プロセッサの台数、データ転送、同期のオーバーヘッドを考慮に入れ、どこまで階層的に分割を行うかを決定することが重要となる。一般的には、図２のようにデータ依存により並列性を抽出しにくい場合や、他のマクロタスクに比べて実行コストが著しく大きなマクロタスクが存在するような場合に、次の階層でのマクロタスクの生成を行う。現在は、この過程を自動化するために、クリティカルパス長と各マクロタスクの実行コストの比を使用することを検討している。

４．まとめ

本稿では、一つのマクロタスクの内側を階層的に分割し、ダイナミックスケジューリングと、スタティックスケジューリングを効果的に組み合わせて階層的なマクロデータフロー処理を行う手法について述べた。

今後の課題としては、プログラムの持つ並列性に合わせて、どのマクロタスクをどの階層まで分割し並列処理を行なうかということを適切に判断する基準の導入と、プログラム中の並列性により、各階層でのプロセッサのクラスタリングを可変とし、コンパイラが自動的に決定する手法の開発が挙げられる。

５．参考文献

[1] 笠原博徳、成田誠之助、橋本親．ＯＳＣＡＲ (Optimally SCheduled Advanced multiprocessoR) のアーキテクチャ．信学論 J71-D(8) PP.1440-1445, 8 1988.

[2] 本多弘樹、岩田雅彦、笠原博徳．Fortran プログラム粗粒度タスク間の並列性検出手法 信学論，J73-D-I(12)，12 1990.

[3] 本多弘樹、広田雅一、笠原博徳．階層型マルチプロセッサシステムＯＳＣＡＲ上でのFortran並列処理手法．並列処理シンポジウムＪＳＰＰ’８９論文集、PP.251-258， 2 1989.

[4] 本多弘樹、水野聡、笠原博徳、成田誠之助．ＯＳＣＡＲ上でのFortranプログラム基本ブロックの並列処理手法．信学論，J73-D-I-(9),PP.756-766, 9 1990.

[5] 本多弘樹、水野聡、広田雅一、笠原博徳．ＯＳＣＡＲ単一プロセッサ・クラスタ上でのFortran並列処理手法．信学技報,CPSY88(33),PP.49-54, 8 1988.

[6] 本多弘樹、入江豊、鈴木真、笠原博徳． ＯＳＣＡＲ上でのFortran並列処理系のインプリメントと性能評価．信学技報，CPSY89,PP.75-80, 8 1989.

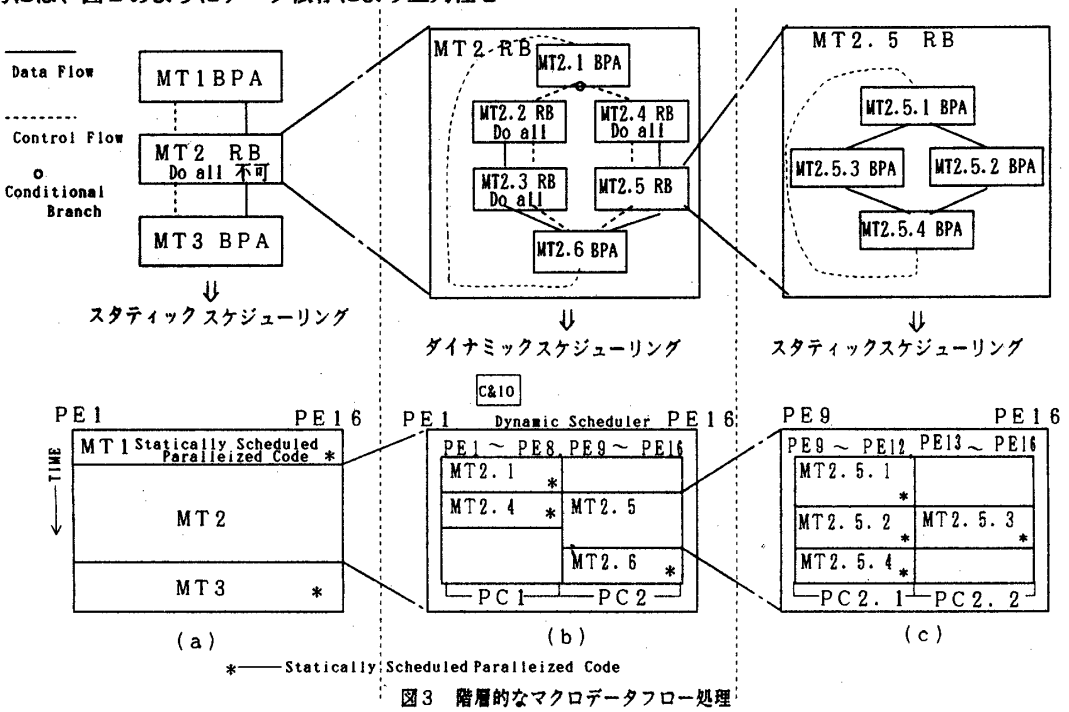


図３ 階層的なマクロデータフロー処理