

4S-1

モジュール分割によるプログラムの
並列性の解析について

菊地重昭 白鳥則郎 宮崎正俊
(東北工大) (東北大通研) (東北大)

1. はじめに

プログラムの大規模化は、並列コンピュータや分散型システムの普及に伴って、Fortran 語等の逐次型高水準言語を用いたプログラムを並列化して処理能力を上げ、より高い応用性を持たせる必要を生じさせた。

筆者等は、プログラムに、主として制御文を境にして分割するモジュール分割法を適用した後、隣接するモジュールの変数間の依存関係を調べて、これを並列化する手法を提案した⁽¹⁾。この並列化手法は、従来の DO 文を中心とした並列化手法の適用に加えて適用可能なものであり、筆者等は、この手法を適用することによってこれまでよりも高いレベルでプログラムを並列化することを目差している。ここでは、この並列化手法を用いて並列化されたプログラムが、複数のプロセッサを持つ並列コンピュータによって処理される時のシステム解析について提案する。

システム解析は、筆者等の提案した並列化手法を既存のプログラムに適用した場合について行ない、従来、主として DO 文を中心に適用してきた並列化と比較検討することを目的としている。また、より能率的なプログラムを構成するために、筆者等の提案した並列化手法を適用したプログラム技法を検討するために利用したい。

2. システムモデル

筆者等の提案する並列化手法により並列化されたプログラムを、複数のプロセッサを持つ並列コンピュータで処理するときのシステムのふるまいを、以下のような待ち行列システムとして記述する。

システムは、図1のように平均入のポアソン到着で到

着するプログラムを受け入れるジョブキューを持つ。ジョブキューで待機しているプログラムは、このキューに接続されている並列化モジュールキュー、および並列化モジュールキューに接続されている並列プロセッサの中にモジュールが全くないときだけ、その並列化モジュールキューに取り込まれる。プログラムは、並列化モジュールキューに取り込まれるとき、モジュール分割による並列化手法が適用され、ここで分割されたモジュールは、無限の長さの並列バッファを無限個持っている並列化モジュールキューにキューの先頭から実行順に並列処理可能なモジュール毎に並ぶ。

並列化モジュールキュー内の並列化されたモジュールは並びの順に並列バッファ内のモジュール毎に並列プロセッサに送り込まれ並列処理される。すなわち、並列プロセッサ内の全てのモジュールの実行が終了すると、並列化モジュールキューの内の次の並列バッファの中の全てのモジュールが同時に並列プロセッサに送り込まれ、並列処理される。このような操作が繰り返し実行されて、並列化モジュールキュー内および並列プロセッサ内の全てのモジュールの処理が終了すると、ジョブキューで待機するプログラムがあれば、そのときに限りその新たなプログラムが並列化モジュールキュー内に取り込み可能となる。そして、上述のように並列化され、並列処理可能なモジュールのグループ毎に並列バッファ内に実行順に並べられる。以下、このようにして同様の操作が繰り返し実行されることになる。なお、ジョブキューの長さ、並列バッファの長さや個数、およびプロセッサの数は境界条件を避けるためここでは無限とした。また、モジュール分割および並列化に要するオーバーヘッドタイムは、ゼロとしている。

3. モジュールの統計的性質

プログラムのシステムへの到着を、ポアソン到着とすることは一般に認められているところだが、前述のようなシステムモデルでシステム解析するに当たっては、幾つかの統計的な性質を把握しておかねばならない。そこで、筆者等は、Fortran 語を用いた数値計算用ライブラリについてモジュール分割による並列化手法を適用して、モジュールの実行時間の分布、モジュールの並列度の分布、すなわち並列処理可能なモジュールの数の分布、さらに並列化モジュールキューの長さの分布、すなわち並列化されたプログ

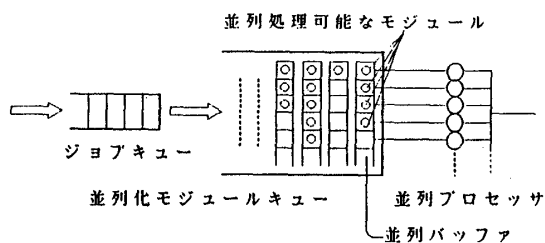


図1. システムモデル

ラムで、並列処理可能な単位でまとめられたモジュールのグループの数の分布等を調べた。これらを、それぞれ図2図3および図4に示す。

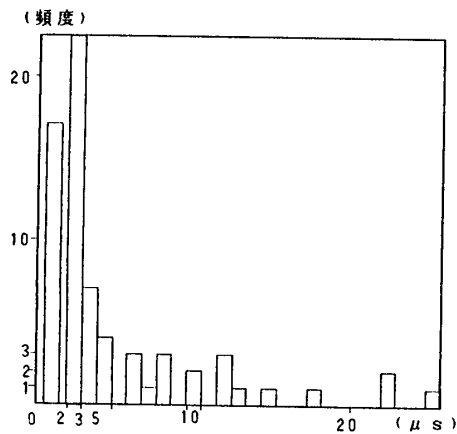


図2 モジュールの実行時間の分布

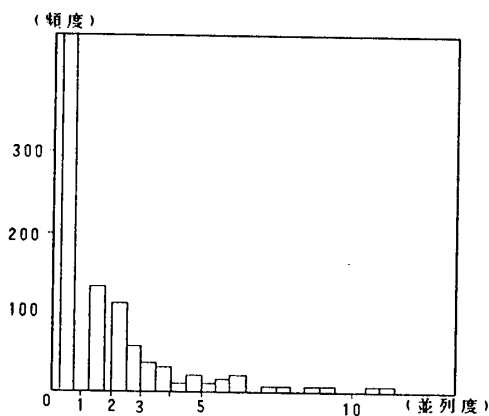


図3 モジュールの並列度の分布

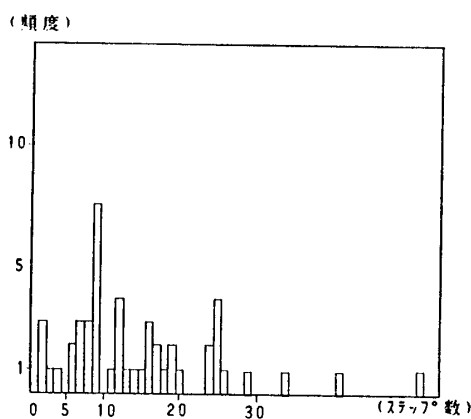


図4 実行ステップ数の分布

これ等の結果より、モジュールの実行時間およびモジュールの並列度は指数分布に近い分布を持っているといえることが分った。

4. システム解析の概要

3. で得られたモジュールの統計的性質を表すデータをもとにして、システムの解析をシミュレーションと待ち行列理論によって行う。以下に、その概要を示す。

筆者等は、数値計算用ライブラリーのソースプログラムについて、筆者等の提案した並列化手法を適用して並列処理したときの並列度の実行順の推移データ、および並列処理したときの実行ステップ数を求めた。シミュレーションによるシステム解析は、これらの実測データを使い、2. で提示したシステムモデルについてコンピュータシミュレーションし、平均システム滞在時間、平均待ち時間、平均ジョブ数、平均待ち行列長等のシステム情報を求める。

また、待ち行列理論によるシステム解析は次のような手順で行う。2. で述べたシステムモデルにおいて、並列プロセッサにより同時に処理されるモジュールの処理時間に関する分布関数または確率密度関数を求める。並列化モジュールキュー内の並列バッファの中のモジュールは、互いに異なる並列バッファ同志では異なる時間帯にプロセッサにより処理されるから、それらの処理時間は独立な事象の中で決定されることになる。従って、プログラムの処理時間の分布関数または確率密度関数は、同時に処理されるモジュールの処理時間の分布関数または確率密度関数のたたみこみによって求めることができる。これよりプログラムの処理時間の平均値と分散が求められる。2. のシステムモデルにおいて、並列化モジュールキューと並列プロセッサをひとつにまとめて、ひとつのプロセッサと考えれば、このシステムモデルは、M/G/1の待ち行列モデルとなり、上で求めた平均値と分散を使って平均システム滞在時間、平均待ち時間、平均ジョブ数、平均待ち行列長等のシステム情報を求めることができる。

5. むすび

筆者等は、文献(1)で提案した並列化手法を適用した並列処理システムのシステム解析を待ち行列理論により行った。この解析結果が、妥当かどうか調べるために現在コンピュータシミュレーションを行っているが、この理論的解析の妥当性が保証されれば、モジュール分割のみを適用して並列化しない場合のシステム解析も理論的に容易に求めることができるから、DO文のみに並列化を適用していた従来の並列化手法との評価の比較が理論的かつ定量的にできることになる。

6. 参考文献

- (1) 菊地、白鳥、宮崎「逐次型高水準言語プログラムのモジュール分割による並列性の抽出について」電情通学論 D, Aug. '88