6B－5

疎結合並列計算機上でのプロセッサ・ファームの効果的実装法

安原　浩春　　梅尾　博司
{haru,umeo}@umeolab.osakac.ac.jp
大阪電気通信大学　工学部

１．はじめに

これまでに様々な並列処理の手法が考案されてきた．その中にプロセッサ・ファームと呼ばれる手法がある[1][2].プロセッサ・ファームは非常にスケーラビリティがありプロセッサ数が変化した時の対応が容易である．このためプロセッサ・ファームの手法をトランスピュータ上で実用化した例も多く紹介されている[3][4][5]．本稿では，プロセッサ・ファームの性能を64台のトランスピュータを用いて，プロセッサ数，タスク数，タスク粒度，並びに負荷のばらつきの３つの点から解析し，より効率的にプロセッサ・ファームを実現する手法を提案する．

２．プロセッサ・ファームとは

プロセッサ・ファームの基本的な動作を説明する．プロセッサ・ファームは１つのコントロール・プロセッサと複数のワーク・プロセッサから構成される．各ワーク・プロセッサはあらかじめ割り当てられているタスクをコントロール・プロセッサからの司令で実行する．タスクが終了するとその結果をコントロール・プロセッサへ返し，次の司令を待つ（図１参照）．コントロール・プロセッサは，どのワーク・プロセッサが駆動中で，どのワーク・プロセッサが司令待ちなのか，またどれだけのタスクを終了したのかを記録している．プロセッサ・ファームでは，コントロール・プロセッサが１つなので従来の逐次的なプログラムからの移行が容易である．またワーク・プロセッサ数が変化したときにも容易に対応出来る．

３．プロセッサ・ファームにおける負荷分散

一般的に並列処理を行なう場合，各プロセッサの負荷の分散をいかにうまく行なうか，という点が重要である．プロセッサ・ファームで最良の負荷分散は，全てのワーク・プロセッサが同時に終了する場合である．

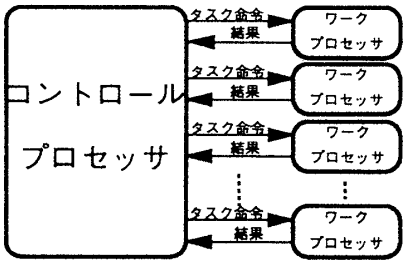

図１　プロセッサ・ファームの構成

しかし，扱うデータの内容によって処理時間が変わるような問題では，全てのワーク・プロセッサが同時に終了するようにうまく問題を分割することは非常に困難である．そこで，問題をワーク・プロセッサ数に対して十分多くのタスクに分割する．そして，プロセッサ・ファームの手法を用いて各ワーク・プロセッサに順番に処理させる．そうすることにより，各タスクの負荷にバラ付きがある場合にも，早くタスクが終了したプロセッサにはすぐに新しいタスクが送られる．これによって，特定のプロセッサだけが早く終了するということがなくなり負荷分散が良くなる．

以上のことから処理時間を特定できないような問題で負荷分散を良くするためには，問題をより多くの仕事に分割すれば良いことがわかる．しかしタスク数を増加させればそれに比例して通信の回数が増加し，通信にかかる時間が無視出来なくなる．より効率の良く処理をさせるためには問題を適当な粒度に分割する必要がある．

４．プロセッサ・ファームの最適化

プロセッサ・ファームをより効率良く使用するためには，タスク数やプロセッサ数，各タスクの負荷のバラ付き等をどのように定めれば良いのか実験した．

Efficient Implementation of Processor Farm on MIMD Distributed-Memory Parallel Computers.
Hiroharu Yasuhara and Hiroshi Umeo
Osaka Electro-Communication University

4.1 プロセッサ数と時間

プロセッサ・ファームの利点の１つに，ワーク・プロセッサの数を簡単に変更できることが挙げられる．図２のグラフは全体として同じ大きさの仕事（ここでは，整数演算を５０万回行った）を様々なプロセッサ数で実行してその実行時間を測定した．縦軸に処理時間，縦軸にプロセッサ数となっている．図２のグラフを見ると非常に奇麗に並列処理の台数効果が現われていることがわかる．

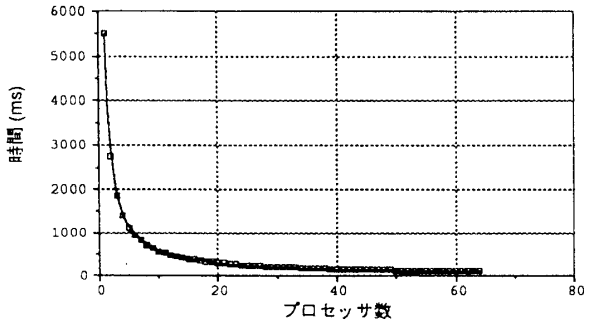

図２　プロセッサ数と時間の関係

4.2 粒度と時間

プロセッサ・ファームでは，全体の処理をどのような粒度に分解するかも重要な要素の一つである．図３は６４台のプロセッサを用いて500,000,000回の実数演算を様々な粒度で行い，その時の時間を測定したグラフである．縦軸が時間を，横軸は各プロセッサが１つのタスクとして行う実数演算の数である．グラフから粒度が，1,000以下では，通信のオーバーヘッドが大きく，1,000,000以上からは，各プロセッサの負荷が大きすぎることが判る．しかし1,000から1,000,000までのかなり広い領域で非常に安定していることが判る．

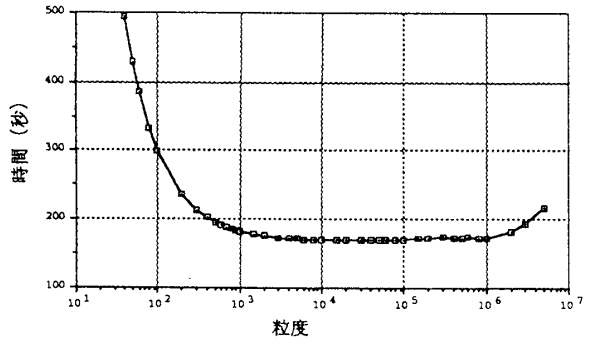

図３　粒度と時間の関係

5．おわりに

本報告では，プロセッサ・ファームの有効性を示すとともに効果的な実装法を示した．本報告ではトランスピュータを用いることで比較的容易にプロセッサ・ファームを実現でき，その成果も十分満足できるものとなった．プロセッサ・ファームは問題の解法が複数の問題に分割できる場合に非常に有効である．また，今回はタスクの分割に主眼をおいたがこの他にも様々な高速技法が考えられる．例えば通信量が大きい場合はワーク・プロセッサにタスクバッファを持たせることで結果を送り出してから次のタスク命令が来るまでの間のアイドル時間を小さくできる．この様なテクニックは他にもまだあるだろう．また，本報告ではトランスピュータを用いたが，他の並列計算機では異なった性質が現われるかもしれない．疎結合並列計算機の場合，通信能力と演算能力とのバランスが全体の性能に大きく影響するからである．少しでも多くの並列計算機で有効な効果的実装法というものを考えるならば，通信能力や演算能力をパラメータとしなければならないだろう．そして，そのような解析を数多く行うことで，通信能力と演算能力とのバランス等の並列計算機の理想的なアーキテクチャが明らかになっていくだろう．

参考文献

[1] Ronald S.Cok, PARALLEL PROGRAMS FOR THE TRANSPUTER, PRENTICE HALL, 1991, pp57-87 （梅尾，他（訳）：”Transputer/Occamによる並列プログラミング入門”，共立出版，（1993年，10月予定））

[2] INMOS, THE TRANSPUTER APPLICATOINS NOTEBOOK Architecture and Software, May, 1989, pp39-46

[3] S.Bakhteyarov, E.Dudnikov, D.Jahn, Oscillatory System Simulation on a Transputer Network, WoTUG newsletter, January, 1992, pp34-38

[4] P.E.Hannington,SENSION GOES DOWN THE TUBES, SERC/DTI TRANSPUTER INITIATIVE MAILSHOT, October, 1991, pp19-20

[5] K.Khachaturjan, S.Safronov, I.Sinitsa, The Kocdoc Operating System.Structre Concept and the Main Operating Principles, SERC/DTI TRANSPUTER INITIATIVE MAILSHOT, June, 1991, pp57-74

[6] 山本正樹，中井泰明，村上安範，トランスピュータ入門，日刊工業新聞，1990

[7] 木村博昭，福島實，トランスピュータによる並列処理，海文堂，1990