

II-204 トランスピュータを用いた並列処理によるFEM計算の効率化

法政大学 正会員 西谷 隆 亘

C R C 正会員 渡辺 隆之

法政大学 学生会員 土屋 直也

東京大学 学生会員 仲江川 敏之

1. はじめに 運動方程式を用いた二次元流出モデルの解法としてFEMを用いると、運動方程式に含まれる非定常・非線形性の故に解を得るには繰り返し計算時間を要し、手軽なパソコンでは手に負えない。このような計算量が極めて多い大規模計算はEWSや大型汎用機などを用いなければ、実用的な時間内で行えない。今日、並列処理はスーパーコンピュータ等で利用されており注目を集めているが、特にパソコンに接続して使用でき、安価に並列処理システムを構築できることから注目を集めているのがトランスピュータで、いくつかの研究がなされている¹⁾。トランスピュータはOCCAM言語のプロセスモデルを効率よく実現するために設計された簡単な命令セットを持つRISCタイプの32ビットマイクロプロセッサである。パソコンに接続できる経済性の高いCPUである²⁾。本論はFEMプログラムをこのトランスピュータを用いて、大次元のFEM計算が実用に耐える時間内で計算できるようにし、他の機種との性能を比較したものである。

2. FEMプログラムの並列化 既存のFEMプログラムの全体のアルゴリズムの流れを変えずに並列化する。既存のFEMプログラムではNewton-Raphson法を用いることにより近似解を求めている。

- ① 近似解の誤差（連立一次方程式の定数項）の計算
- ② 近似解の誤差判定
- ③ Jacobi行列（連立一次方程式の係数行列）の計算
- ④ 近似解の修正値（連立一次方程式）の計算
- ⑤ 近似解の修正

この内、仕事量の多い①、③のプロセスをデータ分割法により並列化した。今回の並列化においては機械的にデータ分割法を適用するもの（以下並列化1）と、プログラマーの判断によりプログラム内で計算量を考慮してデータ分割法を適用するもの（以下並列化2）の2種類を作成した。

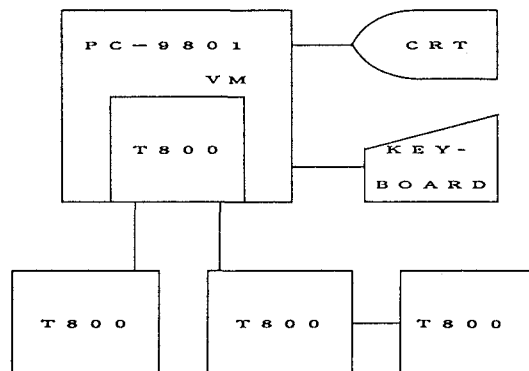


図1 システム構成

パラレルFORTRANでプログラミングする際、割付処理（パラレルアプリケーション）で行った。

3. システム構成と使用言語 (1)ハードウェア トランスピュータは2本のケーブルを接続するだけでCPUの組合せ個数と組合せのコンフィギュレーションは自由にできる。本研究でのシステム構成は図1に示すようパソコンと4個のトランスピュータからなる。これはMIMD型の疎結合マルチプロセッサに分類される。(2)ソフトウェア プログラミングに用いた並列処理言語はパラレルFORTRANである。パラレルFORTRANはOCCAM言語をハネスとして作られた並列処理の記述が可能な言語である。この言語は通常のFORTRANの命令に加えてOCCAMのプロセス概念に基づいたチャンネル通信によるマルチタスクプロセッシングが可能である。

4. 並列処理の性能評価 並列処理時の性能を示すパラメータの一つに速度向上率があり次式で表される。

$$S_n = T_1 / T_n : \text{速度向上率}$$

T_i : i 台のプロセッサで実行したときの時間

n : 使用したプロセッサ台数

解析例として4種の仮想流域での流出解析計算を行った。表1

表1 全体の速度向上率

節点数	10	20	36	54
並列化1	1.01	1.53	1.86	1.85
並列化2	2.02	2.48	2.74	2.52

に節点数と速度向上率の関係を示す。いずれの場合も節点数が36のとき速度向上率は最高値をとり、並列化1の速度向上率は最高で2.0程度、並列化2では最高で2.7程度になっている。並列化の効果は明白である。「同じプログラムであってもデータに対して均等に仕事量を各プロセッサに分担させるプログラムを作成することは極めて困難で、試行錯誤を繰り返してはじめて経験的に異種のデータに対しても仕事量を均等に分割できる」ことが、割付処理では避けては通れない最大の問題点である。この内並列化2の各部分ごとの速度向上率を図2に示す。まず、並列化したプロセス①、③に着目すると、節点数が10から36までは上昇するが節点数36と54では値が増加しなくなる。これは節点数が少ないときには並列化に伴う時間損失(逐次プログラムの並列化に伴うデータ通信や同期など)が各部分の仕事量に比して多いためである。節点数が多くなると時間損失の全体に占める割合が少なくなるので上昇する。節点数36から54にかけて速度向上率が3.0程度で上昇しなくなっているのは割付処理の処理速度がシーケンシャルアプリケーションより低くなるためで、割付処理の処理時間を用いて速度向上率を計算すると、3.8と高い値であった。この値は並列化に伴う時間損失を考慮すれば、この二つの部分の並列化手法は充分良いものと評価できる。次に、全体の速度向上率は節点数が10と20のケースではプロセス①、③の部分だけ見る場合、同じような値と傾向を示すが、節点数が36のとき最大値をとり54の時には逆に低下している。この原因は並列化していないプロセス④の部分の影響が強く、図3のようにプロセス④の部分は節点数が少ない時には無視できる程度であったが節点数が54の時には20%以上を占めるまで増加しているためである。更に節点数の多い仮想流域での計算ではプロセス①、③の速度向上率がこれ以上向上しないことを勘案すれば、プロセス④の部分を並列化しない限り全体の速度向上は望めない。

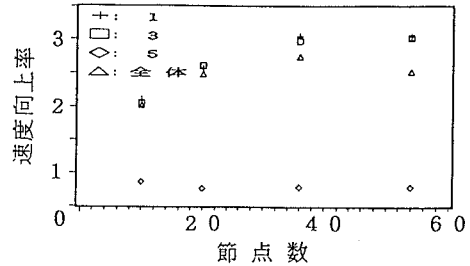


図2 節点数と速度向上率の関係

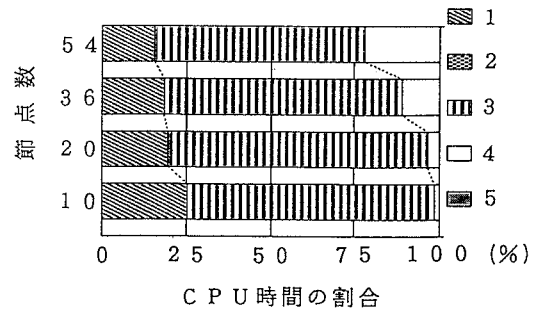


図3 節点数とCPU時間の割合の関係

5. 他の計算機との処理速度の比較 トランスピュータによる並列処理が他の計算機に比べてどの程度の処理能力があるのか検討する。ここでN0.4のデータに対する処理時間をM780/10で処理に要したCPU時間を単位時間としてその何倍程度で処理されているかで示たのが図4である。計算量を考慮した並列プログラムの実行時間はSUN-4/1と同程度、またM780/10のCPU時間と比較しても4倍弱程度で処理できた。

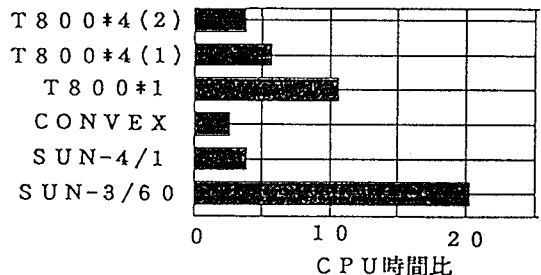


図4 節点数54のときのCPU時間比

6. 結論 並列化された部分については高い速度向上率がえられた。また、全体の処理時間についてもパソコンと比較的安価なトランスピュータとからなるパソコンレベルの並列ネットワークシステムにより、大規模FEM計算が実用に耐える時間内で計算され得ることがわかった。

【参考文献】1) 井門俊治, 石根正久, 久保田昌晴: プラズマシミュレーションにおける並列化の研究, 日本機械学会, 第67期通常総会講演会講演論文集, No. 900-14, pp. 510-512, 1990

2) 山本正樹, 中井泰明, 村上安範: トランスピュータ入門, 日刊工業新聞社, 1990年