4H－7

並列シミュレーションマシン Cenju 上の有限要素法
－ 剛性行列の作成部分の並列化 －

加納　健† 中田 登志之† 奥村 秀人‡ 大竹 邦彦‡ 中村　孝‡ 福田 正大‡ 小池 誠彦†
(† 日本電気(株) C&C システム研究所 ‡ 航空宇宙技術研究所)

1　はじめに

我々は、64台の要素プロセッサで構成された並列シミュレーション・マシン Cenju[1] を開発して来た。並列計算機のアーキテクチャの評価を行なう際には、実用レベルの大規模な応用プログラムが必要であり、その一つとして、非線形の有限要素法の剛性行列の作成の部分について評価したのでそれについてここに報告する。

2　並列シミュレーション・マシン Cenju

並列シミュレーションマシン Cenju は8個のクラスタで構成され、各クラスタ間はパケット交換の多段網で接続されている。各クラスタは、バスで接続された8台の要素プロセッサ(PE)で構成されている。メモリ空間は各 PE に分割された分散共有メモリの構成を採っており、アドレスにより他のプロセッサのメモリをアクセスすることができる。また、各 PE 間の同期は barrier を用いて実現することが可能である。

3　有限要素法

3.1　有限要素法の処理

一般に有限要素法では、対象物の挙動をいくつかの要素の集合体の節点自由度の運動ととらえる。対象物の変位、速度、加速度は、要素の辺上にある節点の自由度方向に対して求められる。

対象物に対して荷重ベクトル $\vec{L}$ が加えられたときの各自由度の変位 $\vec{D}$ は、対象物の剛性行列 K によって関係付けられ、

$$K\vec{D} = \vec{L}$$

となる。非線形問題では、K 、$\vec{L}$ は、$\vec{D}$ の関数であり、近似解 $\vec{D}$ に対して、剛性行列 K と、荷重ベクトル $\vec{L}$ を求め、変位 $\vec{D}$ が収束するまで、連立一次方程式の求解を繰り返す。

衝撃解析等で問題となる有限要素法の非線形構造問題では、Newton-Raphson 法の逐次近似の毎ステップごとに剛性行列を作り直さないと収束が悪い場合がある [2]。1回の Newton-Raphson ループの中では、剛性行列の作成と線形求解が主な処理である。このような解析の場合は、線形解析の有限要素法と異なり、剛性行列の作成部分が計算時間の大きな割合を占める。従って、非線形問題を高速に処理するためには、剛性行列の作成部分と連立一次方程式の求解部分を並列化する必要がある。ただし、今回の報告では前者の剛性行列の作成部分の評価に限って報告する。

剛性行列の作成処理は、要素の種類によって異なった処理を行なわなければならず、ベクトル化が難しい問題である。しかしながら、処理の前半の部分(次節、処理1)は、各要素ごとに、完全に独立に計算でき、並列処理効果が得られやすい問題である。

今回は、米国カリフォルニア大学で開発された構造解析用プログラムである NONSAP[3] の剛性行列作成部分を Cenju 用に並列化した。評価を行なったのは2次元アイソパラメトリック要素の非線形問題で、平面応力のみを生ずるモデルである。要素1266個、節点自由度2743個のデータを用いた。

3.2　剛性行列の作成

剛性行列 K は、各要素の要素剛性行列の足し合わせによって求められる。荷重ベクトル $\vec{L}$ は、外力と、各要素でのひずみによる応力ベクトルとの和から計算される。要素剛性行列、応力ベクトルは、その要素の材質、形状等の情報、要素内節点自由度の変位から計算される。

剛性行列を作成する処理は大きく次の2つに分けられる。

処理1 要素の情報、要素内自由度の変位から要素剛性行列、応力を計算する。

処理2 要素剛性行列、各要素の応力ベクトルを足し込むことにより、剛性行列、荷重ベクトルを計算する。

処理1は、各要素間で並列に計算できる。また、同じ要素特性を持つ要素は、ほぼ同じくらいの処理量となる。

処理2は、並列化が難しい問題で、並列化する場合のボトルネックとなる処理である。

今回の実験では、処理2を1台のプロセッサで行なう方式(並列化1)と複数のプロセッサで行なう処理方式(並列化2)の2つの実験を行なった。

4　並列化1

4.1　並列処理方式

処理1は、複数のプロセッサ(スレーブプロセッサと呼ぶ)で行ない、処理2は1台のプロセッサ(マスタプロセッサと呼ぶ)で行なう。マスタプロセッサのメモリには、各スレーブプロセッサ用に要素剛性行列と応力ベクトルを格納するためのバッファを設ける。処理手順は、以下のようになる。

step1 スレーブプロセッサは、割り当てられた要素の要素剛性行列、応力を計算する。
マスタプロセッサは、1つ前のループで書き込まれたバッファ内の要素剛性行列、応力を該当する剛性行列、荷重ベクトルの要素へ足し込む。

barrier1

step2 スレーブプロセッサは、作成した要素剛性行列、応力をマスタプロセッサのバッファに書き込む。

barrier2

この処理を未処理の要素がなくなるまで繰り返す。

4.2　実験結果

この方式を評価した結果を図1に示す。

本方式では、PE 台数が20台を越えたあたりで、速度向上率の傾きが鈍くなってしまう。これは、使用する PE 台数が増えると、step1 のマスタプロセッサのでの足し込み処理にかかる時間が、各スレーブプロセッサでの要素の処理に比べて大きくなってしまうからである。その結果、全 PE 間での同期をとる barrier1 の終了時刻がマスタプロセッサの足し込み処理の終了によって決まるようになってしまう。(図2参照)

処理1を行なうプロセッサが多くなると、マスタプロセッサだけで行なっていた処理2を並列化しなければならない。

Evaluation of Finite Element Analysis on the Parallel Simulation Machine Cenju – Parallel Calculation of Stiffness Matrix –

Yasushi KANOH†, Toshiyuki NAKATA†, Hidehito OKUMURA‡, Kunihiko OHTAKE‡, Takashi NAKAMURA‡, Masahiro FUKUDA‡, Nobuhiko KOIKE†
†C&C Systems Research Laboratories, NEC Corporation
‡National Aerospace Laboratory

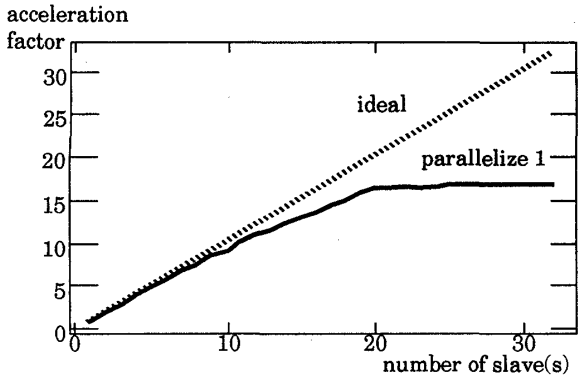

図 1: 並列化 1 の速度向上率

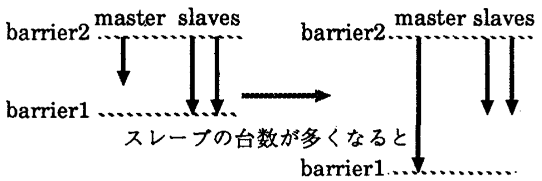

図 2: 速度向上率の飽和の原因

5 並列化 2

作成した剛性行列と荷重ベクトルは、連立一次方程式の求解に使われる。このときは、各節点自由度ごとに処理される。本実験では、連立一次方程式の求解を並列化することを念頭に置き、節点自由度ごとに各 PE に分割格納する。

本節では、各 PE で、処理 2 を並列に行い、剛性行列、荷重ベクトルを分散格納する方式について述べる。

5.1 並列化の問題点

処理 2 を並列に行うには、次のような方式が考えられる。

1. 剛性行列の各要素、荷重ベクトルの各要素に対し、それぞれ、それを作成するときに必要な、要素剛性行列の成分、応力ベクトルの成分のバッファを用意する。
2. 各 PE は、要素剛性行列、応力ベクトルを計算後、該当する剛性行列の成分、荷重ベクトルの成分を格納している PE のそれぞれのバッファに書き込む。
3. すべての要素の処理が終了後、各 PE はバッファ内の値を加算し、割り当てられた剛性行列の成分、荷重ベクトルの成分の値を計算する。

ここで、問題になるのが、ある PE が要素剛性行列成分や応力ベクトルの成分をバッファに書き込むときのバッファ内のアドレスである。同じアドレスに 2 つの値を書き込むと、上書きされ、加算した結果が正しい値にならなくなる。

要素を処理した PE がその成分をバッファのどの位置にすべきかを表す関数 f が必要となる。

本方式では、要素内での各節点自由度に与えられる要素内自由度番号を用いた。(図3参照)

すべての要素での自由度番号の付け方が同じならば、1 つの節点自由度が、それを共有する要素内で呼ばれる自由度番号は、すべて異なる。本方式では、この性質を用いて、要素内自由度番号を関数 f の値としている。

5.2 並列処理方式

step1 各 PE は、要素の要素剛性行列、応力ベクトルを計算し、各成分の該当するバッファを格納している PE のメモリに書き込む。

割り当てられた要素全てに付いてこの処理を繰り返す。

barrier1

step2 各 PE は、バッファ内の値を加え、割り当てられた剛性行列、荷重ベクトルの成分を計算する。

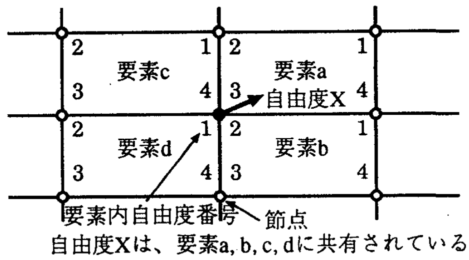

図 3: 要素内自由度番号

5.3 実験結果

図4に本方式での速度向上率のグラフを示す。PE 台数 64 台のときに 1 台のときの約 49 倍の速度向上を示している。

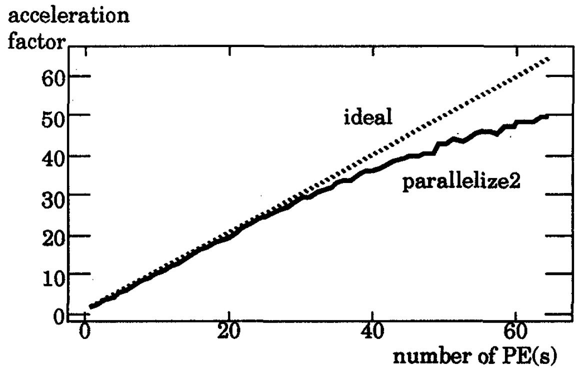

図 4: 並列化 2 の速度向上率

高い速度向上率が得られた要因としては、処理 2 を並列に処理していることの他に、プロセッサ間同期をとる barrier を 1 度しか行なわなくてよいことがあげられる。

一方、本方式には、以下のような問題点もある。

1. 今回評価したプログラムは、2 次元要素に関するものであるが、実用では、3 次元要素も扱われる。3 次元要素の場合には、節点を共有する要素数が増加し、本方式では、バッファの容量の問題が生じる可能性がある。
2. 要素が均質な形状をしていない場合には、関数 f をあらたに定義しなければならない。

6 おわりに

本報告では、並列シミュレーションマシン Cenju の応用の一つとして、有限要素法での剛性行列の作成部分を並列化し、その評価結果について述べた。その結果、並列化 2 のような方式をとれば、プロセッサ数が増大しても台数効果の傾きが鈍くなることなく高速化することがわかった。

有限要素法を並列化によって高速化するには、2 番目に処理時間がかかる連立一次方程式の求解部分の処理を、並列化する必要がある。現在、直接法、反復法の並列化を検討を行なっている。

なお、本研究の一部は、科学技術庁官民特定共同研究の一環として行なった。

参考文献

[1] 中田, 松下, 小池 他: 並列回路シミュレーションマシン *Cenju*, 情処 30 周年記念論文, 情処学会誌 Vol.31, No.5, (1990)

[2] 大竹, 福島, 安永, 山本: 大型衛星フェアリングの分離挙動の数値シミュレーションについて, 日本航空宇宙学会第 31 回構造強度講演集, (1989)

[3] Klaus-Jürgen Bathe, Edward L. Wilson, Robert H. Iding: *NONSAP*, Structural Engineering Laboratory University of California Berkley, UCSESM 74-3, (1974)