14

並列シミュレーションマシンCenju上の有限要素法

中田 登志之* 加納 健* 小池 誠彦*
奥村 秀人** 大竹 邦彦** 中村 孝**
福田 正大**

Evaluation of Finite Element Analysis on the Parallel Simulation Machine Cenju

by

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

ABSTRACT

This paper describes parallelization of Finite Element Methods (FEM) on the Parallel Simulation Machine Cenju. One of the most time consuming problems in FEM which we decided to investigate is nonlinear dynamic finite element analysis. The main loop of the nonlinear dynamic finite element analysis is the loop based on Newton-Raphson method which is composed of the following two stages, namely:

1. Calculation of the Stiffness Matrix
2. Solution of a set of linear equations.

We decided to tackle the above two problems independently. Generation of the stiffness matrix consists of:

Process 1 Per Element Calculation in which the element stiffness matrix as well as the reactance force is calculated for each element.

Process 2 Accumulative Calculation in which the element stiffness matrices are combined to form the global stiffness matrix.

It is quite straightforward to parallelize **Process 1**, just allocate a group of elements to processors and let them calculate the values for the elements. On the other hand, **Process 2** is not so easy to parallelize. We tried two strategies to parallelize **Process 2**. By eliminating the serial bottleneck, we were able to attain a speed-up of 48 on a 64 processor system. For parallelizing solution of linear equations, we tried two approaches.

1. Prallelizing LU-Factorization.
2. Parallelizing Conjugate Gradient Methods.

We first evaluated parallelizing LU-Factorization on a comparatively small set of data. However, due to insufficient amount of parallelism, the speed-up was low-2.9 using 7

* 日本電気(株)C&Cシステム研究所
** 航空宇宙技術研究所

This document is provided by JAXA.

processors. So, we decided to tackle parallelizing the conjugate gradient method. The conjugate gradient method we chose is Scaled Conjugate Gradient Method (SCG) which was developed by Hayami for vector computers. For a large scale matrix (8,904 x 8,904) we were able to attain a speed-up of 36 on a 64 processor system.

1. はじめに

構造関係の解析手法としては有限要素法が広く使用され，確立されているが，この手法は現在市販されているスーパーコンピュータでは，高速処理を行う上で問題があることが明らかになりつつある。このため，有限要素法の並列処理に着目して並列計算機で有限要素法を高速に処理する手法の研究を行い，その実証を試みることとした。その１ステップとして汎用構造解析プログラムNON-SAP[2]を並列化し，日本電気(株)で開発した並列シミュレーションマシンCenju[1]上で評価を行った[4]ので報告する。

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。Cenjuのシステム構成を図１に示す。

3. 有限要素法

3.1 有限要素法の処理

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[3]。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。なお，今回の報告では両者の問題を各々独立な問題として扱う。

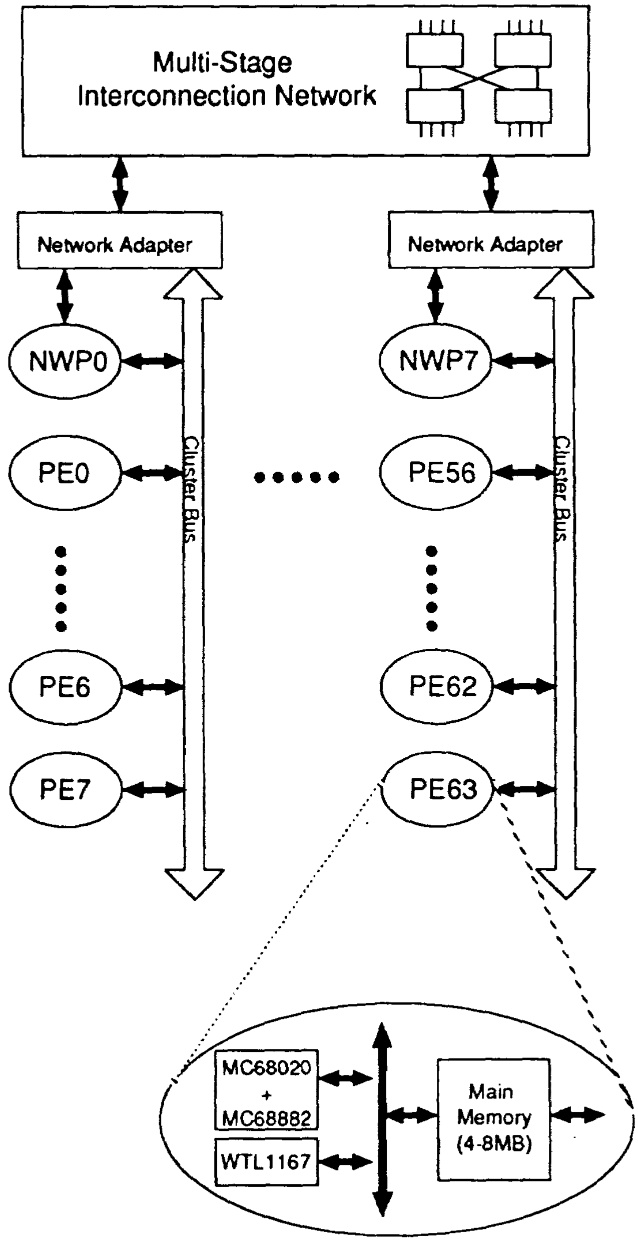

図１ Cenjuのシステム構成

This document is provided by JAXA.

3.2 剛性行列の作成

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。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う方式（並列化１）と複数のプロセッサで行う方式（並列化２）の２つの実験を行った。

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。

step 1

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。

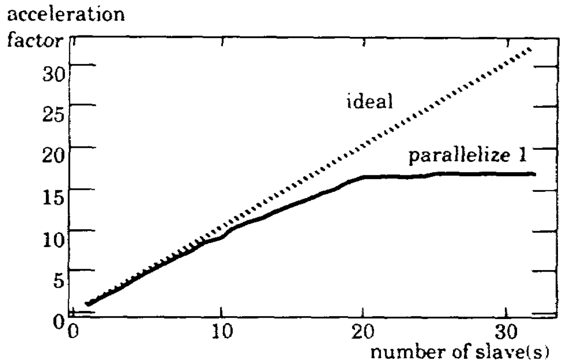

図２　並列化１の速度向上率

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。

barrier 1

step 2

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。

barrier 2

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２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，step 1 のマスタプロセッサ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 barrier 1 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。

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

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

This document is provided by JAXA.

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。

5. 2 並列処理方式

step 1

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。

barrier 1

step 2

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。

5. 3 実験結果

図 3 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。

高い速度向上率が得られた要因としては，処理

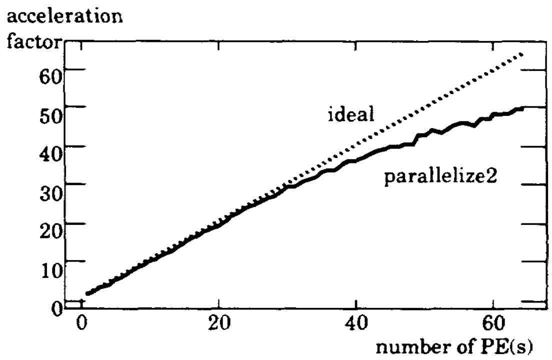

図 3 並列化 2 の速度向上率

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，節点番号とバッファの要素を一意に対応づける方法を決定しなければならない。

6. 線形求解の並列化 -1 LU 分解の直接並列化

一般的に構造解析に用いられる有限要素法では，剛性行列の格納方法としてスカイライン法を用い，線形求解の解法として LU 分解を用いる。スカイライン法を用いた対称行列の LU 分解は a) ピボット行に対する演算（ループ a と呼ぶ）と b) ピボット行を用いた演算（ループ b と呼ぶ）の 2 種類の演算からなる。

LU 分解を並列化するに当たっては，各列毎の要素をプロセッサに割り当てる方法を採用した。

LU 分解で並列化した場合の評価データをループ b だけ並列化した場合，並びにループ a，ループ b 共に並列化した場合の両方について，表 1 に示す。速度比は逐次版のプログラムを 1 としている。いずれの場合にも，2 倍程度の速度向上で飽和している。

これは元々，LU 分解での並列性が基本的に剛

This document is provided by JAXA.

表１　LU 分解の並列化

	ループ b のみ		ループ a,b 共	
台数	実行時間	(速度比)	実行時間	(速度比)
1	367	(0.96)	447	(0.79)
2	229	(1.53)	255	(1.38)
3	191	(1.84)	199	(1.76)
4	176	(1.99)	175	(2.00)
5	172	(2.04)	164	(2.14)
6	173	(2.03)	163	(2.15)
7	179	(1.96)	165	(2.13)

性行列のバンド幅分しかないのに対し，有限要素法では Fill-In の数を減らして処理量を減らすためになるべくバンド幅を小さくしようとする工夫が成されられている事に起因する。

7.　線形求解の並列化 -2 SCG 法の検討

前節の評価から LU 分解だけでは十分な並列性が得られないことが判明した。そこで，反復法の一種である，共役勾配法の並列化を検討することにした。

7.1　SCG 法の概要

SCG (Scaled Conjugate Gradient) 法[5] は速水等によって提案された，対角項によるスケーリングを施した共役勾配法の一種であり，その主な処理は以下に示すようになる。

1. $D^{-1}=diag[1/a_{11},\ 1/a_{22},\ \cdots\ 1/a_{nn}]$ を求める。
2. $r_1=b-Ax_1$
   $p_1=D^{-1}r_1$
3. i に関して反復計算
   - $\alpha_i=\dfrac{(r_i,\ D^{-1}r_i)}{(p_i,\ Ap_i)}$
   - $x_{i+1}=x_i+\alpha_i p_i$
   - $r_{i+1}=r_i-\alpha_i Ap_i$
   - $\beta_i=\dfrac{(r_{i+1},\ D^{-1}r_{i+1})}{(r_i,\ D^{-1}r_i)}$
   - $p_{i+1}=D^{-1}r_{i+1}+\beta_i p_i$

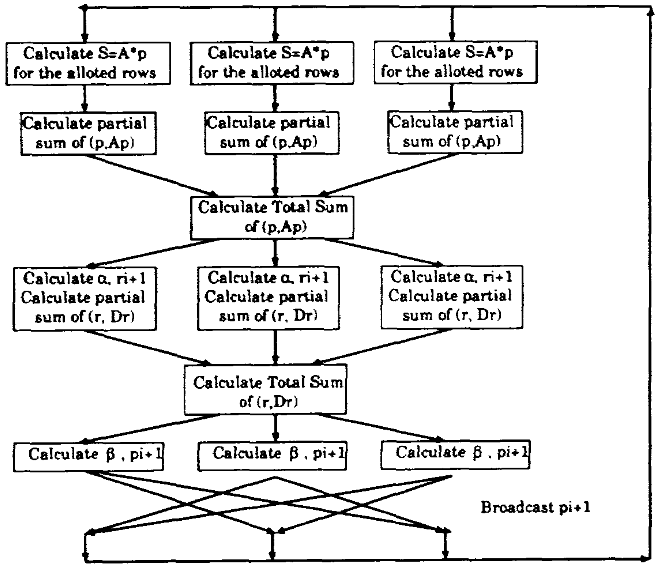

図４　SCG 法の並列アルゴリズム

但し Ap, $D^{-1}r$, 内積などは反復当たり１回計算すれば良い。演算量の大半は Ap の行列×ベクトル→ベクトルの計算に費やされる。

7.2　SCG 法の並列化

SCG 法の並列化に当たっては従来対称行列で上三角形だけを保持していたものを上下三角形両者を保持するようにと，基本的には行毎にプロセッサに割り当てることにより並列化した。この時の並列化アルゴリズムを図４に示す。

この時プロセッサ間通信は以下の２点で発生する。

1). $A\times p=s$ を行うのに必要な p の転送。各プロセッサでは p のベクトルの要素のうち割り当てられた要素に対して新しい値を得た後，担当する要素の値を全プロセッサに伝えないといけない。

2). ベクトル対ベクトルの内積を演算する時に部分和の加算。

後述する様に，1) の転送が深刻なオーバヘッドとなった。

7.3　大規模行列に対する SCG 法の評価 (1)

8904×8904（非零要素184,776個）のデータで評価を行った。その時の速度向上を図５に示す。10 台程度で７倍の速度向上で飽和している。これは，

This document is provided by JAXA.

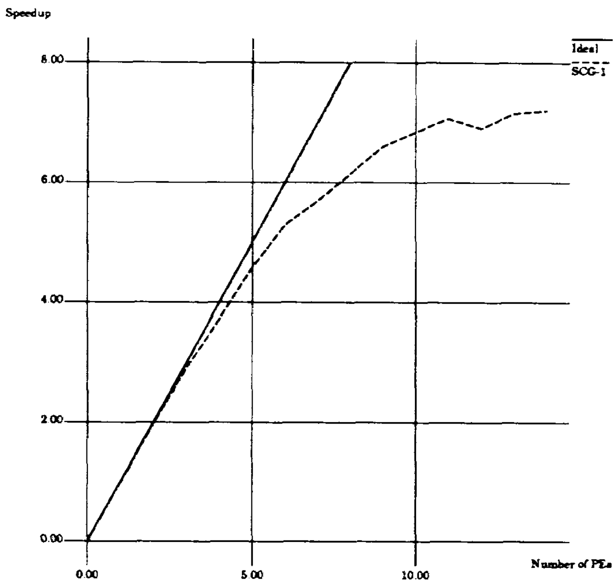

図５　大規模行列に対する速度向上（その１）

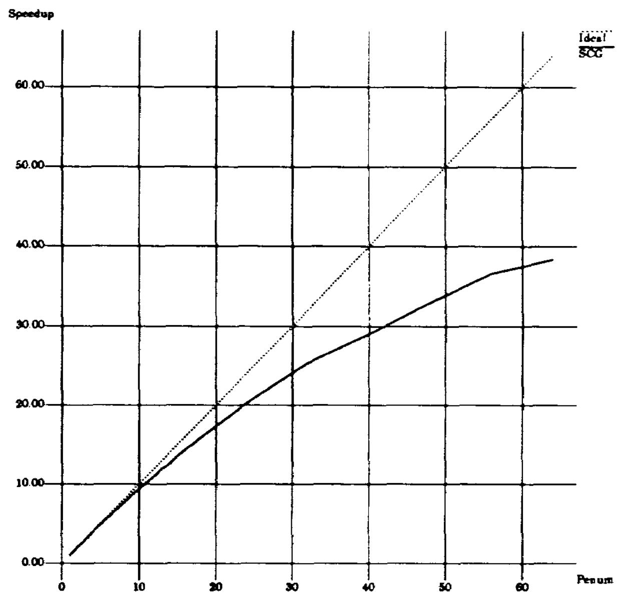

図６　大規模行列における速度向上（その２）

SCG 法で予測していたものと比べて，はるかに低い。そこで，１反復の場合の各演算部分の処理時間を計測した。その結果，18台の時には通信時間の方が$A \times p$に要する時間よりも時間がかかっていることが判明し，これが処理の隘路になっていることが分かった。

7.4　通信量の減少

前節の評価ではpの要素の通信時間が速度向上を妨げる事が分かった。ところで，pのベクトルの各要素が，sに影響を及ぼす範囲はsの全領域ではなく基本的には，Aの行列の形状によって定まる（これはLU分解におけるバンド幅の概念と似ている）。この影響範囲は前もって計算できるので，従って各行の割当を帯状にすると，pの各要素を通信するのはその要素が影響を及ぼす範囲を担当するプロセッサだけですむ。

有限要素法で用いる行列は大体が，バンド幅が小さい行列であるため，通信範囲は小さいと考えられる。

7.5　大規模行列における SCG 法の評価(2)

図６にデータのプロセッサへの割当を変更し，pの各要素は影響を与えるプロセッサにしかデータを転送しないように変更したプログラムでの速度向上を示す。64台の時に１台の時の38倍の速度向上を達成し，線形求解での並列化で，行列生成と同じ程度の速度向上が得られる目処が得られた。

8.　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，有限要素法を並列化し，その評価結果について述べた。その結果，1)剛性行列作成部分に関しては高い並列化効果が得られること。2)線形求解の部分はSCG法などの反復法を用いると高速化できることがわかった。

今後の検討課題としては，３次元要素への対応，バッファ領域の削減，SCG法の収束性の検討などがあげられる。

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った。

参考文献

1)　中田　他：並列回路シミュレーションマシンCenju，情報処理，Vol.31，No.5，pp.593-601，May，1990.
2)　Klaus-Jürgen Bathe，Edward L. Wilson，Robert H. Iding：*NONSAP*，Structural Engineering Laboratory University of California Berkley，UCSESM 74-3，(1974).
3)　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

This document is provided by JAXA.

いて，日本航空宇宙学会第31回構造強度講演集，(1989).

4) 加納 他：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－，情報処理学会第42回全国大会予稿集，pp.6-72-6-73, (1991).

5) 速水，原田：対角項スケーリングを施した共役勾配法のベクトル計算機における有効性について，情報処理学会論文誌，Vol.30, No.11, pp.1364-1375, (1989).

This document is provided by JAXA.

This document is provided by JAXA.