

- 5 大規模有限要素解析における
グリッドコンピューティングのための耐障害性の適用
The Application of Fault-Tolerance to Grid Computing in Finite Element Analysis

大浜 拓也¹ 竹原 和夫² 三上 市藏³
Takuya OHAMA, Kazuo TAKEHARA, and Ichizou MIKAMI

抄録：グリッドコンピューティングは地理的に分散した計算資源をネットワークを介して利用することで大規模な処理能力を実現するための技術である。一方でグリッドコンピューティングはネットワークに接続して演算処理を行うため、計算ノードの故障やネットワークの停止などによる障害性を抱えている。著者らは既にグリッドコンピューティング環境で利用できる有限要素法による並列プログラムを開発してきているが、本研究ではこのアルゴリズムを修正し、さらにグリッドコンピューティングの障害性に対応する手法を考案し、そのプログラムの妥当性と全体の解析時間への影響について検討した。その結果、計算途中から計算ノード数を変更可能であることから、グリッドコンピューティングにおける大規模有限要素解析において、計算ノードに起こりうる障害に対応可能であることがわかった。

Abstract : Grid computing is an innovative technology that obtains computing power by using the computer resource through the network. However, grid computing has the possibility to be suffered some problems which are either hardware or network failure. The parallel computation of finite element analysis has already been achieved by grid computing. In this paper, some techniques on fault tolerance in grid computing was developed on the parallel finite element analysis. Moreover, the accuracy of this method and the influence for the entire computation time was discussed. From the numerical results, it was found that the number of computing nodes can be modified in the middle of the calculation, and that this technique is efficient for the fault-tolerance to grid computing.

キーワード：グリッドコンピューティング, 耐障害性, 有限要素法, Globus Toolkit, MPICH-G2
Keywords : Grid Computing, Fault-tolerance, Finite Element Method, Globus Toolkit, MPICH-G2

1. はじめに

近年ネットワーク技術の向上とコンピュータの飛躍的な性能向上に伴い、地理的に離れた複数の機関の高性能な計算システムを使い、大規模な科学技術計算を実行できる、グリッド^{1),2)}と呼ばれる次世代計算基盤についての研究が行われている³⁾。

グリッドコンピューティングは遠隔地の計算ノードへインターネットを利用してアクセスするため、計算ノードやネットワークの予期せぬ障害に遭遇する可能性が高く、ハードウェア障害や災害による計算機の停止、ソフトウ

アの停止といったさまざまな障害が考えられる⁴⁾。そこで Ninf-G⁵⁾や CyberGRIP⁶⁾ではグリッドコンピューティングの障害に対応した汎用アプリケーションを運用している。これらの汎用アプリケーションでは演算中に障害が発生した場合、障害ノードで処理されている演算を他の計算ノードに振り分けることで対応している。

一方土木分野でも、構造解析はより現実に示しうる挙動に近い結果を得るために、解析モデルのメッシュを細分化し、大規模なものとなってきている⁷⁾。著者らは既にグリッドコンピューティング環境で利用できる有限要素法による並列プログラム⁸⁾を mpi⁹⁾を用いて開発してきている。しか

¹ 学生会員 関西大学大学院 総合情報学研究所

(〒 569-1095 大阪府高槻市霊仙寺 2-1-1 TEL 072-690-3213 E-mail ohama@sc.kutc.kansai-u.ac.jp)

² 正会員 博(工) JIP テクノサイエンス(株) 東京テクノセンター 解析技術部 (〒 103-0025 東京都中央区日本橋茅場町 1-2-5 Tel:03-5614-3204)

³ フェロー 工博 関西大学 工学部都市環境工学科 教授 (〒 564-8680 大阪府吹田市山手町 3-35-35 Tel:06-6368-1121)

しグリッドコンピューティングにおける有限要素法を用いた並列計算では、対象モデルを領域分割し、各計算ノード間で通信を行い計算を行うことになるので、計算途中でどれか1つの計算ノードに障害が発生しただけですべての計算が止まり、始めから計算を行わなければならない。クラスタソフトウェア SCore¹⁰⁾には障害対応として各計算ノードに対してスペアノードを配置する方法や、実行中のプログラムをファイルに退避したり、退避したプログラムを再実行させるチェックポイント・リスタート機能があるが、グリッドコンピューティングにおける有限要素法を用いた並列計算においてはこのような障害対応機能は用意されていないようである。さらに mpi プログラムは計算途中で並列計算台数を変更することは不可能であるが、グリッドコンピューティングを用いた並列計算では障害が発生したノードを外す、すなわち計算途中において並列計算台数を変更する機能も必要である。

本研究では既報の並列プログラム⁸⁾のアルゴリズムを修正し、さらにグリッドコンピューティング環境において耐障害性に適用できるように拡張する。さらにこれを構築したグリッド環境に実装して、並列計算台数変更の効果を検証する。

2. 耐障害性アルゴリズム

(1) 途中経過の保存

耐障害性を考慮した大規模有限要素解析のアルゴリズムとして、反復計算中において各計算ノードで求められる途中情報をサーバへ送信して保存を行う、また障害発生後に保存されたこの途中情報を読み込む。また、計算途中からのリスタートを可能にする。

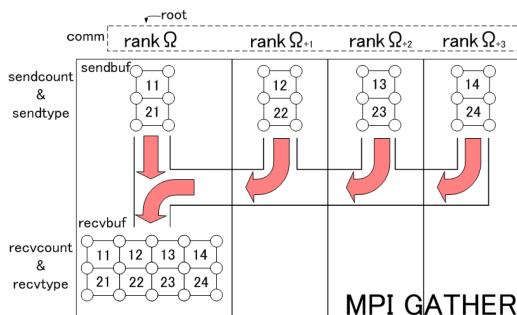


図-1 変位と変位速度のサーバへの送信

保存方法は図-1 のように行い、並列プログラムに以下の mpi サブルーチンを用いることで可能とした。

```
CALL MPI_GATHER(sendbuf, sendcount, sendtype
, recvbuf, recvcnt, recvtype, root, comm, ierror)
```

(2) サードパーティファイル転送機能

計算再開時においてサーバに保存された途中情報を入力データとしてすべての計算ノードに転送しておく必要がある。ここでは Globus Toolkit 標準の GridFTP によるサードパーティファイル機能¹¹⁾を用いる。

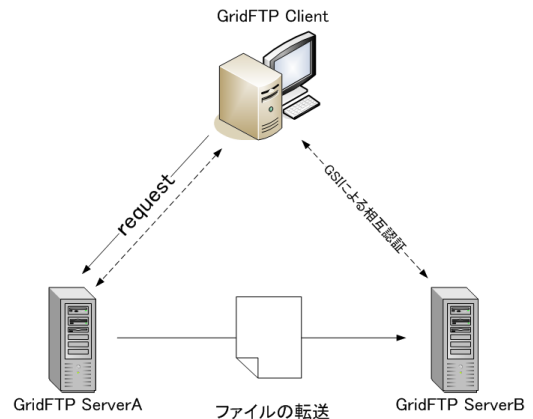


図-2 GridFTP によるサードパーティファイル転送

GridFTP によるサードパーティファイル転送は図-2 のように転送される。従来の FTP の機能では図-2 のようなファイル転送を実行しようとした場合に、一度クライアントがサーバ A から転送したいファイルをダウンロードし、サーバ B にアップロードするという手順を踏まなければならない。しかしクライアントが URL(gsiftp://<ホスト名>/<ファイルのフルパス名>の書式)を指定し、サードパーティファイル転送を行うことで省略することができる。ここではこの機能を用いて、サーバが保持している途中情報を計算再開時に計算可能なノードへ一斉転送する。

(3) リスタート

転送されたデータを各計算ノードで初期値として読み込み、途中状態からの計算を可能とすることができる。

(4) 遊休資源の利用

リスタート時に計算ノード数を減少させると解析時間が長くなってしまいうため、他に計算可能な遊休資源を利用することも可能である。手法として Globus Toolkit における MDS(Monitoring and Discover Service) 機能¹¹⁾を用いる。MDS は標準的なインターフェースとグリッド環境で利用されるリソース情報を定義しており、その情報には、OS 名

やディスクのサイズといった静的なデータ, またはメモリやCPUの使用状況といった動的なデータを持っている. そこで以下のコマンドを実行することでその情報を取得する.

```
%grid-info-search
```

また MDS コマンドに以下のような検索条件を付ける事でグリッド環境に属している複数のホストの中からリモートジョブの必要要件にあったホストを検索することができる.

```
%grid-info-search -h grid.civil.kansai-u.ac.jp
-b "Mds-Vo-name=civil.kansai-u.ac.jp,o=Grid" (&
(Mds-Device-Group-name=processors) (Mds-Cpu-Tot
al-Free-1minX100>=***) (Mds-Cpu-SpeedMHZ>=***))"
Mds-Cpu-Total-Free-1minX100 Mds-Cpu-SpeedMHZ
```

3. 数値計算手法

(1) 数値計算手法

解析領域を図-3 に示すように Ω_m と Ω_{m+1} とする.

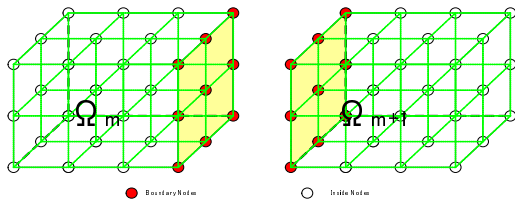


図-3 領域の分割

まず要素に対して有限要素法の考え方¹²⁾から要素剛性マトリクス $[K_e]$ を領域 Ω_m, Ω_{m+1} のそれぞれについて求める.

$$[K_e] = \iiint [B]^T [D] [B] dV \dots\dots\dots (1)$$

ここで $[B]$ はひずみマトリクス, $[D]$ は材料剛性マトリクスである. 次に時刻 $(p-1/2)\Delta t$ において $[K_e]$ に $\{\delta\}_n$ を乗じて, 各要素毎の節点力ベクトル $\{f_{e,int}\}_n$ を領域 Ω_m, Ω_{m+1} のそれぞれについて求める.

$$\{f_{e,int}\}_{n,p-1/2} = [K_e] \{\delta\}_{n,p-1/2} \dots\dots\dots (2)$$

式 (2) を全ての要素に適用してモデル全体の節点力ベクトル $\{f_{g,int}\}_{n,p-1/2}$ を得る.

$$\{f_{g,int}\}_{n,p-1/2,\Omega_m} = \sum_{\Omega_m} \{f_{e,int}\}_{n,p-1/2} \dots\dots\dots (3)$$

$$\{f_{g,int}\}_{n,p-1/2,\Omega_{m+1}} = \sum_{\Omega_{m+1}} \{f_{e,int}\}_{n,p-1/2} \dots\dots\dots (4)$$

次に節点 n のつりあいに慣性項と減衰項を付加すると, 以下の動的つりあい式が得られる.

$$m \frac{\partial \{\delta\}_{n,p-1/2}}{\partial t} + C \{\delta\}_{n,p-1/2} + \{f_{g,int}\}_{n,p-1/2} = \{f_{g,ext}\}_{n,p-1/2} \quad (5)$$

m は質量, C は減衰係数, $\{f_{g,ext}\}$ は外力である. ここで式 (5) の偏微分項を時刻 $(p-1/2)\Delta t$ において差分間隔 $\Delta t/2$ で中央差分表示すると, 次式が得られる.

$$\frac{\partial \{\delta\}_{n,p-1/2}}{\partial t} = \frac{\{\delta\}_{n,p} - \{\delta\}_{n,p-1}}{\Delta t} \dots\dots\dots (6)$$

また時刻 $(p-1/2)\Delta t$ における変位速度 $\{\dot{\delta}\}_n$ を, 時刻 $p\Delta t$ と $(p-1)\Delta t$ での変位速度を用いて次式のように表す.

$$\{\dot{\delta}\}_{n,p-1/2} = \frac{\{\dot{\delta}\}_{n,p} + \{\dot{\delta}\}_{n,p-1}}{2} \dots\dots\dots (7)$$

式 (6) と式 (7) を式 (5) に代入して整理すると, 次の運動方程式が得られる.

$$\{\dot{\delta}\}_{n,p} = \frac{1}{1 + 0.5C \frac{\Delta t}{m}} \left[(1 - 0.5C \frac{\Delta t}{m}) \{\dot{\delta}\}_{n,p-1/2} + \frac{\Delta t}{m} (\{f_{g,ext}\}_{n,p-1/2} - \{f_{g,int}\}_{n,p-1/2}) \right] \quad (8)$$

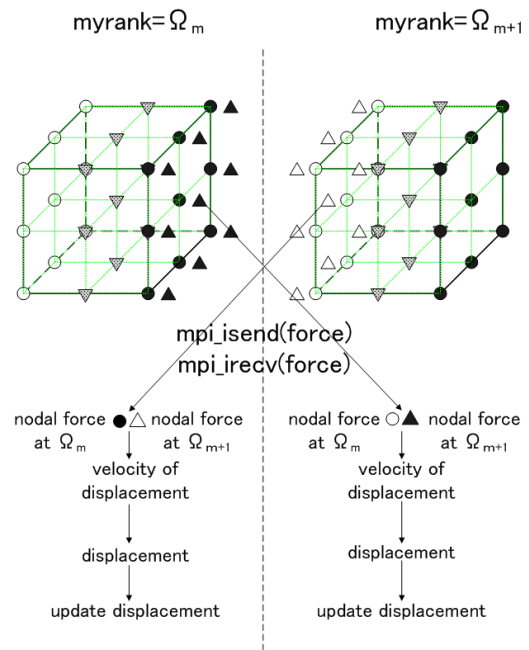


図-4 節点力の送受信

式 (9) は領域 Ω_m と Ω_{m+1} における全ての節点において成り立つが, 境界面における節点力は図-4 のようにそれぞれ隣接する部分領域に通信を行って連続体としての整合性を

保つ。すなわち次式のようになる。

$$\begin{aligned} \{\dot{\delta}\}_{n,p} = & \frac{1}{1 + 0.5C \frac{\Delta t}{m}} \left[(1 - 0.5C \frac{\Delta t}{m}) \{\dot{\delta}\}_{n,p-1/2} \right. \\ & \left. + \frac{\Delta t}{m} \left(\{f_{g,ext}\} - \{f_{g,int}\}_{n,p-1/2,\Omega_m} + \{f_{g,int}\}_{n,p-1/2,\Omega_{m+1}} \right) \right] \quad (9) \end{aligned}$$

ここで通信にはデッドロックを回避するために mpi の非ブロッッキングサブルーチン mpi_isend, mpi_irecv を用いる。

次に結合方程式を求める。変位速度 $\{\dot{\delta}\}$ を時刻 p において中央差分表示すると次式のようになる。

$$\{\dot{\delta}\}_{n,p} = \frac{\partial \{\delta\}_{n,p}}{\partial t} = \frac{\{\delta\}_{n,p+1/2} - \{\delta\}_{n,p-1/2}}{\Delta t} \dots\dots (10)$$

上式を変形すると次式が求まる。

$$\{\delta\}_{n,p+1/2} = \{\delta\}_{n,p-1/2} + \Delta t \{\dot{\delta}\}_{n,p} \dots\dots\dots (11)$$

文献⁸⁾では式(11)の後に $\{\delta\}$ を通信し並列計算を行っていたが、本論文ではこれを改良し、式(9)において $\{f_{g,int}\}$ を通信することとした。

(2) 数値計算手順

数値計算手順を図-5 に示す。

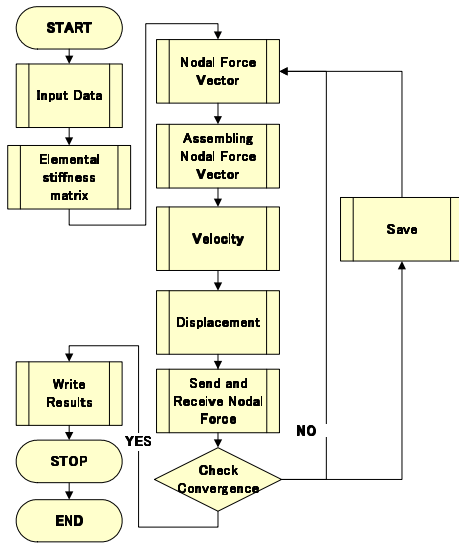


図-5 フローチャート

はじめに入力データとして節点や要素などの構造データと、動的緩和法のパラメータや通信回数などを記したコントロールファイルを読みこむ。各プロセッサは担当する領域を計算し、該当範囲の配列を作成する。次に式(1)より、要素剛性マトリクスを作成する。計算開始時には変位、変位速度を共に初期値を0とする。式(2)、式(3)、式(9)、式(11)より節点ごとの変位と変位速度を求め、隣接する部分領域の整合性を保つために、各計算ノード間で通

信を行い、各領域ごとに境界条件による変位と変位速度の更新を行う。

途中経過を保存する場合は、ある反復回数において式(9)より得られる $\{\dot{\delta}\}$ と式(11)より得られる $\{\delta\}$ を図-1の方法でサーバに保存する。保存された $\{\dot{\delta}\}$ と $\{\delta\}$ は、障害発生後のリスタート時において GridFTP を用いて MDS で検索された計算可能なノードへ送信し、これらを初期値として読み込んで計算を再開する。解の収束判定を行い、収束判定誤差内に収まるまで反復計算を行う。解の収束判定が収束判定誤差内に収まれば解が求まったと判断し、変位、ひずみ、応力を出力し、計算を終了する。

4. 数値計算結果

(1) グリッドコンピューティング環境

計算に用いた仮想グリッドコンピューティング環境⁸⁾のスペックの一部を表-1 に示す。認証を行うサーバに Linux を用い、5 台のクライアントのうち Linux のクライアントを 3 台、Windows のクライアントを 2 台とした。本論文における計算には全て Linux のクライアントを用いた。

表-1 グリッド環境のスペック

Host	Grid01	Grid02 ~ Grid04
Role	Server	Client
CPU	PentiumIV(3.4GHz)	PentiumIV(2.8GHz)
RAM	2GB	1GB
HDD	160GB	80GB
OS	Redhat Enterprise ES	Redhat Professional WS
Network	Gigabit Ethernet	

グリッド環境を構築するために The Globus Project が開発した Globus Toolkit¹¹⁾をミドルウェアとして用い、また並列計算ツールに GlobusToolkit の管理機能やセキュリティ機能を使用して、MPI(Message Passing Interface) アプリケーションを実行させることができる MPICH-G2¹³⁾を用いた。

(2) 解析モデル

実構造物への適用例として、解析モデルを図-6 に示す。L=79.0(m), B=6.5(m), h=3.0~4.9(m) とし、すべて 8 節点ソリッド要素を用いてモデル化した。節点数は 9568, 要素数は 7392 である。材料特性は弾性係数 3.1×10^4 (N/mm²), ポアソン比 0.167 とし、解析パラメータには減衰係数 $C = 1000$, 時間間隔 $\Delta t = 1.0 \times 10^{-4}$, 収束判定誤差 $\Delta = 1.0 \times 10^{-8}$ を用いた。

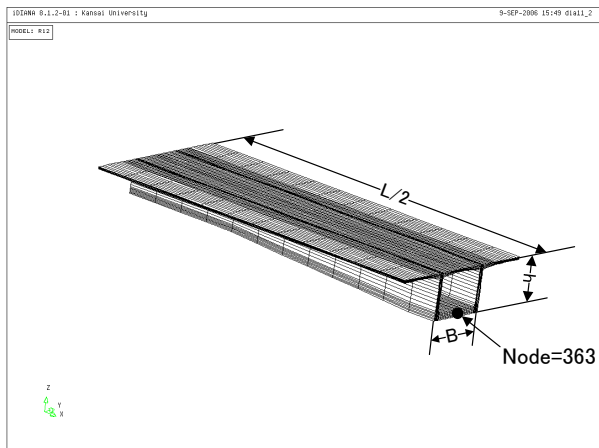


図-6 解析モデル

境界条件としてモデル両側の節点是对称条件として x 方向を拘束し、モデル右下の節点について y および z 方向を拘束した。荷重条件は単位体積重量による自重を考慮した。並列計算のための領域分割は x 方向に分割する方法とし、並列台数は 2~4 台とした。

(3) 解の検証



図-7 本解析手法による変形図

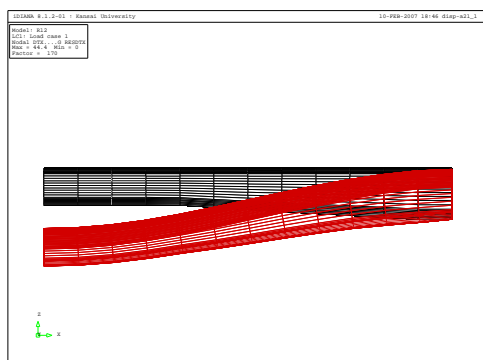


図-8 DIANA¹⁴⁾による変形図

本手法による変形図を図-7に、汎用解析ソフト DIANA¹⁴⁾による変形図を図-8に示す。さらに、図-9に図-6にお

る節点 363 の z 方向の変位量と反復回数との関係を示す。本手法(計算ノード 1 台, single)は繰り返し計算により解が求まり、繰り返し回数 8430 回で変位量-34.67(mm)が得られ、これは DIANA による結果と一致することを確認した。以上のことから、本手法により正しい解が得られていることがわかる。

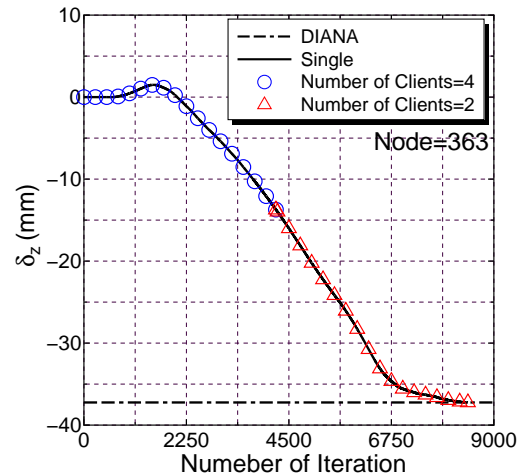


図-9 計算ノード数の変更による解の収束状況

また図-9には台数を 4 台から 2 台に変更した結果も併記している。このときの変更するタイミングは計算ノード 1 台による反復回数の 1/2, すなわち 4215 回とした。図-9より、本手法は計算ノード数 1 台で実行した場合と同様の収束状況を示していることがわかる。よって本手法はリスタートによる台数変化, すなわち有限要素解析における領域分割の変更に対応することができる。

(4) 解析時間

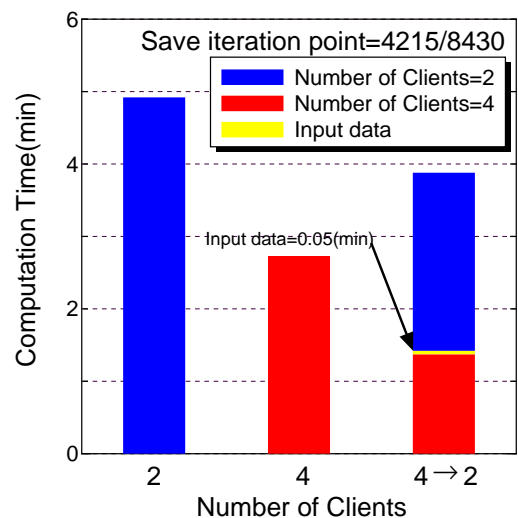


図-10 計算ノード数の変更による解析時間

次に計算ノード数を変更した際における計算時間を図-10に示す。計算ノード数を変更した場合はデータの読み込みの時間が別途必要となるが、その割合は極めて小さい。これより台数の減少による全体の解析時間に少し増加が見られるものの、リスタートによる解析時間の影響は小さいことがわかる。

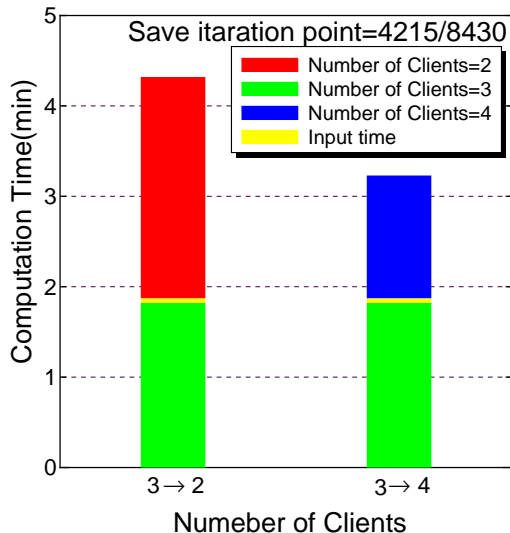


図-11 計算ノード数の増減による解析時間の比較

次にリスタート時に計算ノード数の増減による計算時間の比較を図-11に示す。これよりMDSを利用し、計算ノードを増加させることで全体の計算時間の短縮が可能となり、MDSの利用による有効資源の活用が有効であることがわかる。

5. おわりに

本研究では、グリッドコンピューティングにおける大規模有限要素解析において計算ノードに起こりうる障害に対応する手法を検討した。繰り返し計算により解が求まる本手法のアルゴリズムに着目し、各計算ノードの所有する途中情報をサーバに保存することで、グリッドコンピューティングの障害に対応することが可能となった。本研究で得られた結果より、計算途中からの再計算を行っても解の信頼性を実証し、さらに障害性に対応することも示した。また本手法を用いると単なる障害対応のみならず、弾塑性や有限変形などの非線形解析のリスタートをグリッド環境で実現できると考えられるが、これについては今後の課題としたい。

参考文献

- 1) 日本アイ・ピー・エムシステムズエンジニアリング(株)：グリッドコンピューティングとは何か，Globus Toolkit ではじめるグリッドの基礎，ソフトバンクパブリッシング，2004.4.
- 2) 溝口文雄：グリッドコンピューティング 情報処理の新しい基盤技術，岩波書店，2005.1.
- 3) 谷村勇輔：アジア太平洋グリッド計算の実用化試験，産総研 TODAY，2005.7.
- 4) 武宮博，田中良夫，中田秀基，関口智嗣：大規模長時間実行 Grid アプリケーションの実装と評価先進的計算基盤システムシンポジウム SACSIS 2006，pp.351-358，2006
- 5) 谷村勇輔，池上努，中田秀基，田中良夫，関口智嗣：耐障害性を考慮した Ninf-G アプリケーションの実装と評価ハイパフォーマンスコンピューティングと計算科学シンポジウム，HPCS2005，pp.99-106，2005
- 6) SystemwalkerCyberGRIP
<http://systemwalker.fujitsu.com/jp/cybergrip/>
- 7) 土木工学における計算力学手法の新展開 (SACOMCE) 講習会テキスト，土木学会応用力学委員会，同計算力学小委員会，2005.8.
- 8) 竹原和夫，宮辻和宏，三上市藏，大浜拓也：動的緩和法を用いた有限要素解析のためのグリッドコンピューティングの適用，土木情報利用技術論文集，Vol.31，pp.199-206，2006.10.
- 9) 青山幸也：並列プログラミング入門 MPI 版，理化学研究所 情報基盤センター，2004.3.
- 10) PC Cluster Consortium
<http://www.pcluster.org/>
- 11) The Globus Alliance
<http://www.globus.org/>
- 12) 鷲津久一郎，宮本博，山田嘉昭，山本善之，川井忠彦：有限要素法ハンドブック I 基礎編，培風館，1981.9.
- 13) MPICH-G2
<http://www3.niu.edu/mpi/>
- 14) TNO Building and Construction Research，DIANA User's Manual Release 8.1，2002.

(2007.5.18受付)