

A-025

## CD ブート Linux を用いたグリッドによる FDTD 並列計算の負荷分散特性

## Characteristics of Load Balancing for FDTD Parallel Computation on Grid Computing with Bootable Linux CD

木村 功児<sup>†</sup>  
Koji Kimura園田 潤<sup>‡</sup>  
Jun Sonoda

## 1. はじめに

近年の無線通信の高周波化や UWB (Ultra Wide Band) の民間利用への開放などの無線通信システムの高度化にともない、通信システム最適設計のための電波伝搬解析が重要視されている。近年広く用いられている電波伝搬解析法に Maxwell の方程式を時間領域で計算する FDTD (Finite-Difference Time-Domain) 法 [1] がある。FDTD 法は電磁界の過渡応答解析が可能なが、多くのメモリを必要とし計算時間がかかる問題がある。この計算時間の問題について、並列型スーパーコンピュータ [2]、PC クラスタ [3] 等のさまざまな並列計算アーキテクチャにより解決する研究が報告されている。しかしながら、地理的に離れた場所にある計算機を複数台使用することにより並列計算環境を構築するグリッドコンピューティング(グリッド)を用いた FDTD 並列計算は報告されていない。グリッドには、より多くの計算資源を得ることができ、大規模な並列計算環境を低コストで構築できる利点がある。しかしながら、異機種種の計算機を用いるため各計算機の性能ごとに負荷分散を行う必要がある。さらに、各ネットワークでセキュリティポリシーが異なるためセキュリティが脆弱になりやすい。

そこで本研究では、CD ブート Linux によるグリッド環境の構築と、グリッドにおける FDTD 並列計算の負荷分散の方法として、使用する各計算機の処理速度に応じて複数ジョブを割り当てる方法を適用する。本報告では、CD ブート Linux により構築したグリッドにおける FDTD 並列計算の負荷分散特性について述べる。

## 2. CD ブート Linux によるグリッドの構築

## 2.1 グリッド用 CD ブート Linux の作成と評価

グリッドを構成する各計算ホストに CD ブート Linux を用いることにより、(1) 使用する計算ホストすべてを同じ OS や計算環境に統一できる、(2) 最新のパッケージで CD ブート Linux を作成するためセキュリティを高

めることができる、(3) CD で起動するため使用する計算機のハードディスクを保護できる等の利点があり、グリッドの問題点であるプラットフォームの相違、セキュリティの脆弱性を解決できる。

本研究では、CD ブート Linux を Red Hat Linux 9 をベースに、セキュリティ対策として最新のパッケージにアップデートしたソフトウェア群から作成した。グリッドに不要なパッケージを削除することで、容量は CD 1 枚に収まる 465MB、メモリ使用量は 56MB まで削減した。これにより搭載メモリの少ない計算機でも使用可能となった。また、セキュリティホールも Nessus による評価の結果 0 であった。

## 2.2 CD ブート Linux によるグリッドの構築

図 1 に作成した CD ブート Linux を用いて構築したグリッドの構成を示す。図 1 (a) は、3 つのネットワークにある 3 種類の計算機を 2 台ずつ  $P_1$  から  $P_6$  の計 6 台を用いて構成したグリッド 1 である。図 1 (b) は、5 つのネットワーク内のすべて機種異なる計算機計 12 台を用いて構成したグリッド 2 である。表 1 にグリッド 1、2 を構成する各計算機のハードウェアの仕様を示す。なお、並列計算において計算機間の通信を行うライブラリとして、フリーの MPI (Message Passing Interface) の実装である mpich-1.2.5 を用いた。

## 3. グリッド上の FDTD 並列計算の負荷分散

## 3.1 FDTD 並列計算の負荷分散法

本研究では、FDTD 並列計算の負荷分散法として、解析領域を任意の整数で等分割し複数のジョブを発生させ、各計算機の処理速度に応じてジョブを分配する方法を考える。複数のジョブを発生させることで 1 ジョブ当たりの解析領域を小さくし、処理速度が速い計算機には多くのジョブを、処理速度が遅い計算機には少ないジョブを与える。これにより、全体の計算時間を短縮できる。ここで、各計算機の CPU クロック値を基準にする方法(負荷分散 A)と、FDTD 計算を各計算機で実行した計算時間を基準にする方法(負荷分散 B)を考える。

<sup>†</sup>仙台電波工業高等専門学校 専攻科電子システム工学専攻<sup>‡</sup>仙台電波工業高等専門学校 電子工学科

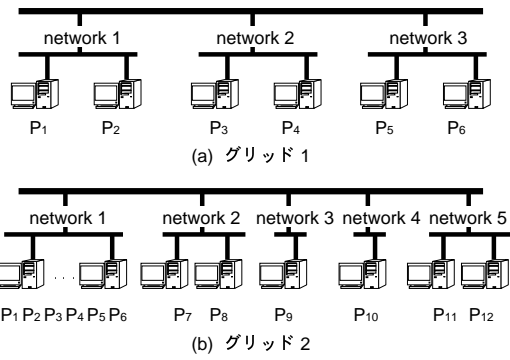


図 1: CD ブート Linux を用いて構築したグリッド

表 1: グリッドを構成する計算機の仕様

(a) グリッド 1

PC	CPU [Hz]	Cache [KB]	Memory [MB]
$P_1, P_2$	Celeron 900M	128	128
$P_3, P_4$	Duron 1.3G	64	128
$P_5, P_6$	Pentium4 1.8G	512	512

(b) グリッド 2

PC	CPU [Hz]	Cache [KB]	Memory [MB]
$P_1$	Celeron 900M	128	192
$P_2$	Duron 1.3G	64	128
$P_3$	Celeron 2G	128	768
$P_4$	Celeron 1G	128	256
$P_5$	Pentium4 2.4G	512	512
$P_6$	Celeron 560M	128	128
$P_7$	Athlon 1.5G	256	256
$P_8$	Pentium3 1G	256	256
$P_9$	Pentium4 1.8G	512	512
$P_{10}$	Pentium3 450M	256	512
$P_{11}$	Celeron 2.5G	128	512
$P_{12}$	Celeron 2.2G	128	512

表 2: 本手法による各計算機へのジョブ割り当て結果

case	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$	$P_7$	$P_8$	$P_9$	$P_{10}$	$P_{11}$	$P_{12}$
a	2	2	-	-	3	3	-	-	5	5	-	-
b	1	1	-	-	3	3	-	-	6	6	-	-
c	1	1	2	1	2	0	1	1	2	0	2	2
d	1	1	2	0	3	0	2	1	2	0	2	1

表 3: 負荷分散を適用した FDTD 並列計算の計算時間

case	without load balance [s]	with load balance [s]
a	28.60	18.00
b	28.60	16.29
c	44.32	24.06
d	44.32	14.17

### 3.2 FDTD 並列計算の負荷分散特性

図 1 および表 1 のグリッド 1, 2 において, 次の case a, b, c, d を考える. case a はグリッド 1 で負荷分散 A, case b はグリッド 1 で負荷分散 B, case c はグリッド 2 で負荷分散 A, case d はグリッド 2 で負荷分散 B である. 解析モデルは, 中心に電流源 (パルス幅 2.5ns のガウシアンパルス) がある自由空間の電波伝搬問題とし,

境界条件には Mur の吸収境界条件を用いた. 解析モデルの大きさは  $600 \times 600$  とした. 表 2 に負荷分散 A, B の手法により得られた各計算機のジョブ割り当て結果を示す. 表 2 より, case c, d はすべて異なる計算機で構成されるため, case a, b に比べ複雑な割り当て数になっている. また, case c, d では割り当て数が 0 の計算機があるが, これは処理速度が遅い計算機を計算ホストから排除した結果である.

表 3 に負荷分散を行わない FDTD 並列計算の計算時間と, 表 2 で示したジョブ割り当て結果を用いて負荷分散を行った FDTD 並列計算の計算時間を示す. 表 3 より, いずれの場合においても計算時間の短縮がはかられており, 計算時間の短縮率は case a では 37.0%, case b は 43.0%, case c は 45.7%, case d は 68.0%であった. 特に, FDTD 計算時間を基準にした負荷分散 B は CPU クロック値を基準にした負荷分散 A よりも計算時間が短縮された. これらのことから, FDTD 並列計算の負荷分散法として各計算機の処理速度を基準に負荷分散を行うことは有効である.

### 4. まとめ

本報告では, CD ブート Linux を用いたグリッド構築とグリッドにおける FDTD 並列計算の負荷分散法を提案した. 作成した CD ブート Linux を評価した結果, 容量 465MB, メモリ使用量は約 56MB, セキュリティホールは 0 個であった. 実際に CD ブート Linux を用いてグリッドを構築し, FDTD 並列計算の負荷分散を行った結果, CPU クロック値を基準にした負荷分散 A, FDTD 計算時間を基準にした負荷分散 B とともに計算時間の短縮がみられた. 特に, 負荷分散 B は負荷分散 A よりも計算時間が短縮され, 負荷分散 B では最大 68% の計算時間短縮がはかられた.

今後の課題として, さらに大規模なグリッド上での実証実験や, 通信遅延も考慮した負荷分散法の提案等があげられる.

### 参考文献

- [1] K. S. Yee, "Numerical Solution of Initial Boundary Value Problems Involving Maxwell's Equations in Isotropic Media," *IEEE Trans. Antennas and Propagation*, 14, 2, pp. 302-307 (1966-5)
- [2] C. Guiffaut and K. Mahdjoubi, "A Parallel FDTD Algorithm Using the MPI Library," *IEEE Antennas and Propagation Magazine*, 43, 2, pp.94-103 (2001-4)
- [3] 園田, "ヘテロ PC クラスタにおける FDTD 並列計算の自動最適負荷分散," 信学論 B, J87B, no. 5, pp.760-764 (2004-5)