

A-005

PC クラスタを用いた FDTD 法電磁界解析の並列化 Parallel FDTD Electromagnetic Distribution using a PC Cluster

佐藤 裕幸†
Hiroyuki Sato

深沢 徹†
Toru Fukasawa

1. はじめに

FDTD (Finite Difference Time Domain) 法[1]は、電界、磁界を時間領域で交互に計算する電磁界解析手法である。ある座標における界の計算は、他の座標の界の計算に影響を与えないので、ベクトル化、並列化が容易な手法であり、多くの並列化研究がなされている[2] - [8]。これまで当社では、CRAY 等の商用並列計算機を用いてベクトル化、自動並列化により計算を行っていたが、今回、安価で拡張性に優れた PC クラスタを用いて、FDTD の並列化を行ったので報告する。

2. FDTD プログラムの並列化方式

FDTD プログラムでは、3次元空間の各座標の電界ベクトルの各 XYZ 成分 (EX, EY, EZ) 及び磁界ベクトルの各 XYZ 成分 (HX, HY, HZ) を以下のような式により計算している。

$$EX1_{xyz} = CEXE_{xyz} \times EX_{xyz} + CEXYH_{xyz} \times (HZ_{xyz} - HZ_{x(y-1)z}) - CEXZH_{xyz} \times (HY_{xyz} - HY_{xy(z-1)}) \quad \dots(1)$$

$$EY1_{xyz} = CEYE_{xyz} \times EY_{xyz} + CEYZH_{xyz} \times (HX_{xyz} - HX_{xy(z-1)}) - CEYXH_{xyz} \times (HZ_{xyz} - HZ_{x(y-1)z}) \quad \dots(2)$$

$$EZ1_{xyz} = CEZE_{xyz} \times EZ_{xyz} + CEZXH_{xyz} \times (HY_{xyz} - HY_{x(y-1)z}) - CEZYH_{xyz} \times (HX_{xyz} - HX_{x(y-1)z}) \quad \dots(3)$$

$$HX1_{xyz} = CHXH_{xyz} \times HX_{xyz} - CHXYE_{xyz} \times (EZ_{x(y+1)z} - EZ_{xyz}) + CHXZE_{xyz} \times (EY_{xy(z+1)} - EY_{xyz}) \quad \dots(4)$$

$$HY1_{xyz} = CHYH_{xyz} \times HY_{xyz} - CHYZE_{xyz} \times (EX_{xy(z+1)} - EX_{xyz}) + CHYXE_{xyz} \times (EZ_{(x+1)yz} - EZ_{xyz}) \quad \dots(5)$$

$$HZ1_{xyz} = CHZH_{xyz} \times HZ_{xyz} - CHZXE_{xyz} \times (EY_{(x+1)yz} - EY_{xyz}) + CHZYE_{xyz} \times (EX_{x(y+1)z} - EX_{xyz}) \quad \dots(6)$$

ここで、例えば、E 1 は現在算出しようとしている電界ベクトルで、E は一時刻前に算出された計算値である。また、C は定数係数 (3次元空間における媒質定数) で計算過程では値は変化しない。この式から分かるように、各座標の値は、一時刻前の計算値を基に算出される。

並列化においては、電界及び磁界の3次元空間領域を分割することにより、各プロセッサは担当領域のみを上記の計算式により処理する。上記の計算式から分かるように、計算においては、計算位置の隣接するデータを使用している。従って、1シミュレーション時刻毎に分割された境界領域のデータを隣接プロセッサ同士で交換する必要がある。今回は、通信するデータが連続して格納されているようにするために、X方向の1次元で領域を分割することにした。X方向で隣接するデータを使用しているのは、計算式より、電界計算においては $HZ_{x(y-1)z}$ と $HY_{x(y-1)z}$ であり、磁界計算においては $EZ_{(x+1)yz}$ と $EY_{(x+1)yz}$ である。すなわち、FDTD の1シミュレーション毎の並列計算は、それぞれのプロセッサで以下のように行われる。

(a) 電界ベクトルの EX1, EY1, EZ1 を X 座標については担当範囲を YZ 座標については全範囲について式(1) ~

(3)により計算する。

(b) 先頭以外のプロセッサは、EY 及び EZ の YZ 平面の先頭のデータを1つ前の隣接プロセッサに送信する。

(c) 最後以外のプロセッサは、EY 及び EZ の YZ 平面の最後のデータを1つ後の隣接プロセッサから受信する。

(d) 磁界ベクトルの HX1, HY1, HZ1 を X 座標については担当範囲を YZ 座標については全範囲について式(4) ~ (6)により計算する。

(e) 最後以外のプロセッサは、HY 及び HZ の YZ 平面の最後のデータを1つ後の隣接プロセッサに送信する。

(f) 先頭以外のプロセッサは、HY 及び HZ の YZ 平面の先頭のデータを1つ前の隣接プロセッサから受信する。

3. 評価実験

今回並列化した FDTD を評価するために、以下の処理環境及び処理方式の実行時間を計測し、性能を比較する。

- 従来の CRAY (J916SV1)を用いた自動並列化による実行を計測し、PC クラスタとの比較を行う。プログラムは、上記式の電界ベクトル及び磁界ベクトルの計算がそれぞれ XYZ 座標の3重ループの形をしている。自動並列化においては、恐らく最外ループを分割することで行われていると考えられる。
- 低速な PC と高速な PC による実行を計測し、CPU 速度とネットワーク速度の比率の違いによる並列化効果の比較を行う。表1にそれぞれのマシン仕様を示す。
- 通信ライブラリ MPI (MPICH-1.2.3) と PVM (PVM-3.4.4) を用いた場合の実行を計測し、通信ライブラリの違いによる性能・並列化効果の比較を行う。一般に、MPI は多くの並列マシンで実装されており、TCP/IP を用いない高速な実装研究も多くなされている。しかし今回は、TCP/IP を用いた実装版を用いており、MPI の方が機能が豊富な分、若干性能が落ちると予想される。
- データ通信において、先に通信すべきデータ部分を計算した後、その部分を送信し、残りの部分を計算した後、データを受信する方式の実行を計測し、(a) ~ (f) の手順方式と比較する。データ送信を先に行うことにより、その他の部分の計算とデータ通信を同時に行うことができ、効率化できることが予想される。

上記は、全て $128 \times 128 \times 128$ セルの立方体領域を1,000時刻シミュレーションした実行時間を計測した。表2に1プロセッサから8プロセッサまで使用した場合の計

表1 計測マシンの仕様

項目	低速PC	高速PC
CPU	Pentium III 933MHz	Pentium IV 2.8GHz
メモリ	SDRAM 512MB	デュアルチャネルDDR SDRAM 2GB
ネットワーク	ファストイーサ	ギガイーサ
NIC	3COM 3c905C-TX	NETGEAR GA622T
OS	Vine Linux Kernel 2.4.19	Vine Linux Kernel 2.4.18

†三菱電機(株) 情報技術総合研究所, MELCO

表 2 計測結果 (単位:秒)

マシン	ライブラリ	方式	1PE	2PE	4PE	8PE
CRAY	—	—	419.00	237.00	132.00	90.00
低速PC	MPI	(i)	1,146.75	620.53	372.54	230.38
		(ii)	1,154.01	609.88	351.84	211.54
	PVM	(i)	1,154.63	615.97	338.83	195.22
		(ii)	1,154.46	630.95	348.32	210.70
高速PC	MPI	(i)	160.52	127.91	123.04	99.75
		(ii)	159.87	101.57	89.24	85.49
	PVM	(i)	159.06	88.79	53.72	33.46
		(ii)	159.32	83.06	45.62	26.28

測結果を示す。また、各マシン及び処理方式における台数効果 (1 プロセッサでの実行時間に対する速度向上率) を図 1 に示す。ここで、方式の(i)(ii)とは、以下の通りである。

- (i) 前記の(a)~(f)の手順で処理した方式
- (ii) 隣接プロセッサに通信すべきデータ部分を先に計算して送信する方式

この計測結果から、以下のようなことが言える。

- CRAY の性能は、1 プロセッサでは低速 PC と高速 PC の間であるが、並列化効果は 8 プロセッサで頭打ちとなっている。これは、自動並列化コンパイラによる並列化効果が 8 プロセッサ程度で限界になっているからだと思われる。一方、我々の並列化では、8 プロセッサを超えても効果が持続している。
- 高速 PC は低速 PC に比べて 1 プロセッサで 7 倍以上高速であり、クロック速度比率をはるかに上回っている。これは、メモリアクセス性能の違い (デュアルチャネル DDR SDRAM 対 SDRAM) だと考えられ、本プログラムはメモリアクセスがボトルネックとなっていることが伺える。
- 低速 PC も高速 PC も MPI より PVM の方が性能が良い。これは MPI の方が機能が豊富な分、各処理が重くなっているためであると思われる。
- 高速 PC で MPI を使用した場合は、ほとんど並列化性能が出ていない。別途計測したレイテンシーやスループット性能はある程度出ているので、MPI の性能がプロセッサの性能に対して低いためであると考えられる。
- 低速 PC + PVM 以外では、(ii)の効果が現れており、特に高速 PC + PVM ではその効果が大きい。

4. おわりに

以上、安価で拡張性に優れた PC クラスタを用いて FDTD プログラムを並列化し、従来使用していた CRAY に比べて 4 倍近い高速化が計れた。これは、システム価格で 1 億円以上のものから約 300 万円と数十分の一になっている。また、プロセッサ数を増やすことによる台数効果は、10 台で 7 倍弱の効果が得られた。更に、CRAY の自動並列化コンパイラでは 8 台程度で効果が頭打ちになっていたが、我々の並列化においてはその傾向がなく、更にプロセッサ数を増やすことにより、効果が持続すると考える。通信ライブラリによる実行性能の違いは、TCP/IP を用いた実装においては、PVM の方が機能が少ない分、効率的であることが分かった。

本件は通信・放送機構の平成 14 年度委託研究「第 4 世代移動体通信システム実現のための研究開発」の成果である。

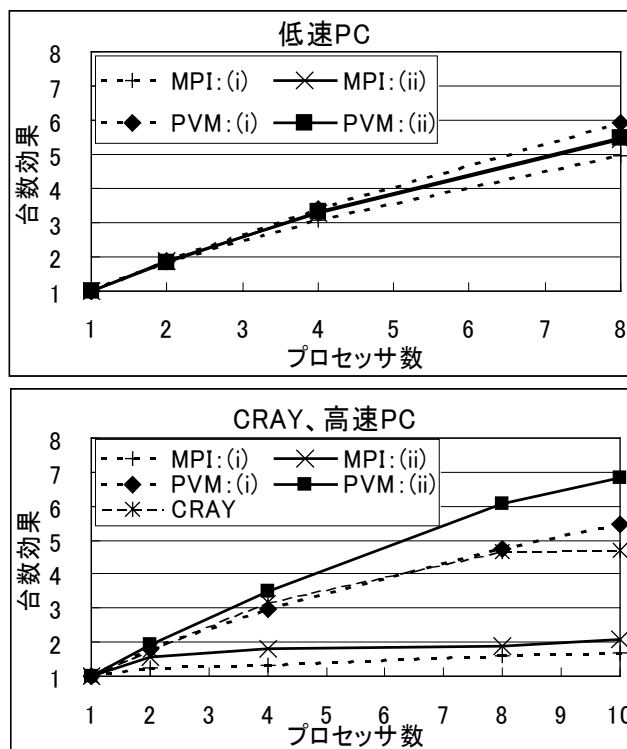


図 1 各マシン及び処理方式における台数効果

参考文献

- [1] K. S. Yee, "Numerical Solution of initial boundary value problem involving Maxwell's equations in isotropic media," IEEE Trans. Antennas & Propag., vol. AP-14, no. 3, pp. 302-307, 1996.
- [2] K. C. Chew and V. F. Fusco, "A parallel implementation of the finite difference time-domain algorithm," Int. J. Numerical Modeling Electronic Networks, Devices and Fields, vol. 8, pp. 293-299, 1995.
- [3] D. P. Rodohan, S. R. Saunders and R. J. Glover, "A distributed implementation of the finite difference time-domain(FDTD) method," Int. J. Numerical Modeling Electronic Networks, Devices and Fields, vol. 8, pp. 283-291, 1995.
- [4] 高田直樹, 安藤勝規, 本島邦行, 伊藤智義, 上崎省吾, "新たな分散 FD-TD 法アルゴリズム," 信学論 C-I, vol. j80-C-I, no.2, pp. 47-54, 1997.
- [5] 服部準, 阿部真, 韓青, "格子状 PC クラスタにおける 3 次元分散 FDTD 法の実装と評価," 情処学 HPC 研報, 69-6, pp.31-36, 1997.
- [6] 打矢匡, 柏達也, "並列型スーパーコンピュータを用いた FDTD 並列計算," 信学論 C, vol. j84-C, no.11, pp. 1122-1125, 2001.
- [7] 大森大介, 石川拓臣, 大友隆, 谷藤忠敏, "FDTD 法を用いた光 CT 像再生の並列計算による高速処理法の検討," 信学技報, OCS-2002-57, PS2002-41, OFT2002-38, pp.31-34, 2002.
- [8] 田口健治, 打矢匡, 柏達也, "大規模電磁界解析を目的とした FDTD 並列計算," 信学技報, MW-2002-133, pp.33-38, 2002.