J. Chem. Software, Vol. 6, No. 2, p. 75–84 (2000)

一次元ポリマーのエネルギーバンド計算における並列処理(2)

寺前 裕之

ATR 環境適応通信研究所, 〒 619-0288 京都府相楽郡精華町光台 2-2
e-mail: teramae@acr.atr.co.jp

(Received: December 3, 1999; Accepted for publication: February 28, 2000; Published on Web: May 1, 2000)

TCGMSG および MPI メッセージパッセンジャーを用いた、一次元ポリマーのエネルギーバンド計算の並列処理について述べる。計算機として、IBM RS6000 4 台によるワークステーションクラスターならびに Pentium II を使用したパーソナルコンピューター 2 台からなる PC クラスターを使用した。以前の報告で使用した二電子積分の並列化では、16 並列とした場合に負荷分散がうまくいっていなかったが、新しいアルゴリズムにより改善できることを示した。Celeron を CPU として使用した PC クラスターでの計算も試みたが、Pentium II よりは並列度が上がらない結果が得られた。MPI を用いた場合には TCGMSG に比べて、メモリー転送のオーバーヘッドが生じるため、やや計算時間が増加する。また実際の計算に要する実時間はネットワークの構造にかなり依存することが判明した。

キーワード : Parallel processing, Crystal orbital, Energy band, PC cluster

1 はじめに

一次元ポリマーに関する非経験的結晶軌道法による計算はポリアセチレンなどの導電性高分子の電子状態研究でその有効性が示されたが、CPU 占有時間およびファイルの入出力共に非常に大規模な計算となるうえ、ベクトル型のスーパーコンピューター向きの計算では無いために、単位セルが小さなポリマー以外の計算はあまりなされていない [1–3]。

近年、単一 CPU でのコンピューターの処理能力も限界に近くなりつつあり、スーパーコンピューター (SC) やワークステーション (WS) も以前ほどの短期間での飛躍的な性能向上は見込めなくなってきた。そこで、現在では複数の CPU を装備した並列処理システムを用いて並列計算を行なうのが大規模計算では主流になりつつあり、SC や WS もクラスター構成ないしは並列型の構成となる。またポリマー計算のように大容量のファイル入出力を伴う場合にはこのファイル入出力待ち時間も大きくなるため並列化することが望ましい。しかし、SC は高価であり、WS レベルにおいても CPU を 20-30 台の単位で確保することは以前と比べて価格が下がってきたとは言え、コスト的に見て難しい。

一方で比較的安価な Intel プラットフォームのパーソナルコンピューター (PC) は近年の性能向上がめざましく、WS との差が縮小してきた。例えば Pentium II 450MHz の Linpack 実測値 (n=100) では浮動小数点演算速度は最高約 90Mflops にも達し、やや古いモデルとの比較ではあるが RS6000/590 の 35Mflops や Cray T932 の単一 CPU での 90Mflops と比較してもむしろ高速な値となっている。また近年、Linux や FreeBSD のような PC で動作するフリーな Unix が普及しはじめたため、PC を WS のように使用する事が可能になり、WS から最小限のプログラム書き換えで PC への移植が行えるようになった。

我々は以前の論文 [4] で一次元ポリマーのエネルギーバンド計算プログラムをソケット通信を用いた TCGMSG メッセージパッセンジャーを利用して並列計算用に書き直しパフォーマンスの計測などを行ない、PC クラスターを用いた並列計算でスーパーコンピューターに匹敵する性能が得られる事を示した。ただし、並列度が増えた場合には、分散処理がうまくいかず、理論値通りに性能の向上が見られず、16CPU で 9 倍程度の加速率に止まった。また PC クラスターは、スピードがかなり異なる CPU を組み合わせていたので、性能向上が計測データには顕著には現れていなかった。そこで本研究では、Intel Pentium II を使用した PC クラスターにより、一層の高速化を目指して並列化に工夫を行い、より並列度を上げる事ができたので報告する。また Pentium II の廉価版である Celeron を使用した結果についても報告する。

本研究では前報 [4] でメッセージパッセンジャーとして採用した TCGMSG がやや古くなりメンテナンスも行われていない現状も考慮して、他に近年並列化ライブラリの標準となりつつある、MPI 環境下での並列化も試み、TCGMSG との比較を行った。さらに、CPU 時間の短縮だけではなく実際の計算に要する実時間についても考察を行ったので併せて報告する。

2 計算方法

ポリマーのエネルギーバンドの計算理論については文献に詳しいので並列計算に必要な部分のみの記述にとどめる [1, 2]。一次元ポリマーの計算理論である結晶軌道法は有限の分子系における分子軌道法を無限系のバンド計算に拡張したものである。

Hartree-Fock 方程式は、

$$\boldsymbol{\varepsilon}(\mathbf{k})\mathbf{S}(\mathbf{k})\mathbf{C}(\mathbf{k}) \;=\; \mathbf{F}(\mathbf{k})\mathbf{C}(\mathbf{k}) \tag{1}$$

である。$F_{rs}(\mathbf{k})$, $H_{rs}(\mathbf{k})$, $S_{rs}(\mathbf{k})$ は、$\mathbf{k}$ をあらわに含まない実空間での各行列要素のフーリエ変換で表すことができて、

$$F_{rs}(\mathbf{k}) = \sum_{j=-N}^{N} \exp(i\mathbf{k}\mathbf{a}j) F_{rs}^{0j} \tag{2}$$

$$H_{rs}(\mathbf{k}) = \sum_{j=-N}^{N} \exp(i\mathbf{k}\mathbf{a}j) H_{rs}^{0j} \tag{3}$$

$$S_{rs}(\mathbf{k}) = \sum_{j=-N}^{N} \exp(i\mathbf{k}\mathbf{a}j) S_{rs}^{0j} \tag{4}$$

ここで、

$$S_{rs}^{0j} = \int \chi_r^0 \chi_s^j d\mathbf{r} \equiv \langle r^0 | s^j \rangle \tag{5}$$

$$H_{rs}^{0j} = -\frac{1}{2}\langle r^0 | \Delta | s^j \rangle - \sum_{h=-N}^{N} \sum_{A}^{atom} \langle r^0 | \frac{Z_A}{|\mathbf{r} - h\mathbf{a} - \mathbf{R}_A|} | s^j \rangle \tag{6}$$

$$F_{rs}^{0j} = H_{rs}^{0j} + \sum_{h=-N}^{N} \sum_{l=-N}^{N} \sum_{t}^{n} \sum_{u}^{n} P_{tu}^{hl} \left\{ 2\langle r^0 s^j | t^h u^l \rangle - \langle r^0 t^h | s^j u^l \rangle \right\} \tag{7}$$

$$P_{tu}^{hl} = \frac{\mathbf{a}}{\pi} \int_{\mathbf{BZ}} \sum_{n}^{occupied} \exp\{-i\mathbf{k}\mathbf{a}(h-l)\} C_{tn}^*(\mathbf{k}) C_{un}(\mathbf{k}) d\mathbf{k} \tag{8}$$

これらの関係式からユニットセル当りの全エネルギーは、

$$\frac{E_{total}}{N} = -\frac{1}{2} \sum_{j=-N}^{N} \sum_{r}^{n} \sum_{s}^{n} \left(H_{rs}^{0j} + F_{rs}^{0j} \right) P_{rs}^{0j} + \frac{1}{2} \sum_{h=-N}^{N} \sum_{A}^{atom} \sum_{B}^{atom} \frac{Z_A Z_B}{|\mathbf{R}_A - \mathbf{R}_B - j\mathbf{a}|} \tag{9}$$

ここで、n は基底関数の数を N は考慮する隣接セル数を表す。式 (7) より容易にわかるように、二電子積分の数は N^3n^4 に比例する。Fock の行列要素を計算するのに必要な電子密度行列は結晶軌道の係数 $C_{tn}(\mathbf{k})$ から計算されるが、結晶軌道の係数は変分方程式を解かないと得ることができない。従って、分子軌道計算の場合と全く同様に SCF 計算が必要である。

Figure 1(a) に示したように、二電子積分および二電子積分の核座標に関する微分の計算において隣接セル数である N を用いた並列化を前報では行っていたが、本研究では (b) で示したようにさらに内側のループへ並列化を移動することにより処理の一層の分散を図った。ただしプログラミング自体は (b) の方が難しくなる。さらに後段の Fock 行列の生成 (式 (7)) において、データの gather をブロック転送のみでは行えなくなるためノード間での和をとる必要が生じる。このため後に述べるように MPI ではオーバーヘッドが生じ、効率が悪くなる。TCGMSG では問題は生じない。二電子積分および二電子積分の核座標に関する微分が計算の全体に占める割合はテストに用いたテフロンポリマーで約 75%であるが、結果として SCF 計算部分も並列処理されるため、並列化される計算量は 90%以上になる。

WS クラスターとしては、通常の 10BaseT イーサネットに接続された IBM の RS6000 590 が 4 台から構成されるものを使用した。PC クラスターとしては、Pentium II 450MHz をデュアル CPU マザーボードを用いて 2CPU 構成としたものを使用した。厳密には PC クラスターではない。PC 用の OS は FreeBSD Version 3.1 の SMP カーネルを使用した。主記憶容量は WS/PC 共に 1 台あたり 256MB である。並列計算を行なうためのライブラリとしては、WS/PC 共に TCGMSG ライブラリ [5] を使用した。また PC 用には MPI の実装の一つである LAM version 6.2β[6] も用いて TCGMSG と比較した。

速度比較のために前報と同様に poly-tetrafluoro-ethylene$(C_2F_4)_x$ を対象に選んだ。基底関数系は STO-3G[8] を用いた。(n=30) 隣接セル数 N は 5 として、二電子積分のカットオフ法には Namur Cutoff 法を用いた [10]。波数ベクトルのサンプリングは 41 点で行ない、Simpson の公式を用い

```
...
ncb=0
DO J= 0,N
  DO K=-N+J,N
    DO L=-N+J,N
      ncb=ncb+1
      if(mod(ncb,nproc).eq.me) then
      DO R=1,n
        DO S=1,n
          DO T=1,n
            DO U=1,n
              Calculate <R(0)S(J)|T(K)U(L)>
                 or     <R'(0)S(J)|T(K)U(L)>
            ENDDO
          ENDDO
        ENDDO
      ENDDO
      endif
    ENDDO
  ENDDO
ENDDO
...
```

(a) 二電子積分およびその微分の並列化（旧）

```
...
icount=0
DO J= 0,N
  DO K=-N+J,N
    DO L=-N+J,N
      DO R=1,n
        DO S=1,n
          DO T=1,n
            DO U=1,n
            icount=icount+1
            if(mod(icount,nproc).eq.me) then
               Calculate <R(0)S(J)|T(K)U(L)>
                  or     <R'(0)S(J)|T(K)U(L)>
            endif
           ENDDO
          ENDDO
        ENDDO
      ENDDO
    ENDDO
  ENDDO
ENDDO
...
```

(b) 二電子積分およびその微分の並列化（新）

Figure 1. 並列計算の概要

て式 (8) の積分を行なった [7]。実時間比較のために、poly-tetrafluoro-ethylene では計算規模が小さすぎるため、poly-(para-phenylene sulfide) $(C_6H_4SC_6H_4S)_x$ を新たに対象に選び、SCF 計算のみを行って CPU 時間ではなく実際に計算が終了するまでの時間を計測した。ここでは 3-21G 基底関数系 [9] を使用し (n=150)、隣接セル数 N は 3、波数ベクトルのサンプリングは 21 点とした。poly-(para-phenylene sulfide) の構造を Figure 2 に示した。

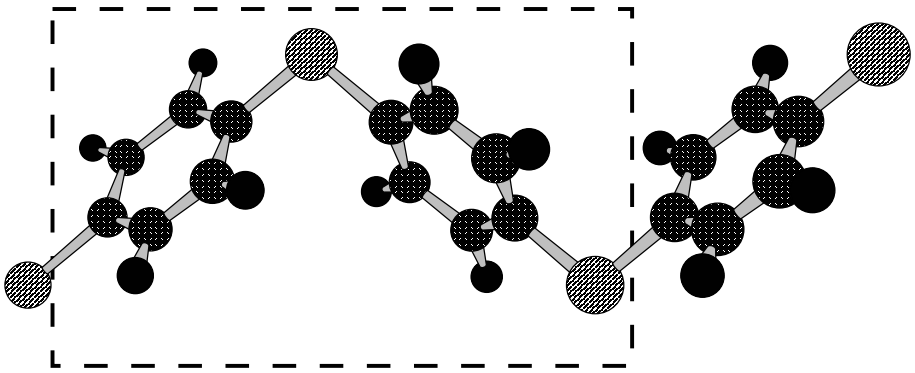

Figure 2. poly-(para-phenylene-sulfide) の構造、破線内がユニットセル

3 結果と考察

Table 1 に計測された CPU 占有時間 (秒単位) を示す。二電子積分およびその核座標に関する微分を旧アルゴリズム並びに新アルゴリズムにより並列化したものについて示した。ただし Celeron プロセサおよび MPI を用いた並列化では新アルゴリズムによるもののみを示した。

ここで並列計算における CPU 占有時間は並列計算時に最も CPU 時間を消費したプロセスの CPU 時間である。速度の計測は各 10 回行ない、最速値をとった。WS および PC クラスターはそれぞれ 4CPU および 2CPU しか持たないため、各 CPU について 1、2、4、または 8 プロセスずつ発生させて、それぞれ 4、8、16 並列計算の値をシミュレートした。ただし同一 CPU 上のプロセス間通信ではセマフォを用いた共有メモリーが使用されるので、純粋に 8CPU、16CPU でソケット通信のみを使用した場合に比べて速度的に有利になっている可能性は残る。ここで加速率を 1CPU 使用での計算時の CPU 占有時間を並列計算時の CPU 占有時間で除算したものとして定義すると、WS の 16 並列計算時では 1 割程度、PC クラスターでは 2 割以上の向上があり、新しい並列化の方法が有効であることを示している。

なお加速率では同等の結果が得られているが、MPI を用いた並列化では、CPU の占有時間が TCGMSG に比べて増加している。これは以下に示すように、ライブラリの仕様の違いによる。つまり各ノードでの計算結果の和を取る場合に、TCGMSG では DGOP ルーチンにより、配列 ARRAY に直接結果を返す事が可能だが、

```
CALL DGOP(MSGDBL,ARRAY,LENGTH,'+')
```

MPI では MPI_ALLREDUCE ルーチンを用いると、

Table 1. Poly-tetrafluoro-ethylene の並列計算の速度比較（秒）

プロセス数	時間 (秒)[a]	加速率	時間 (秒)[b]	加速率
RS6000/590 cluster TCGMSG				
1	342.8	1.00	342.8	1.00
2	182.8	1.88	179.1	1.91
4	88.2	3.89	95.5	3.59
8	52.9	6.48	53.9	6.35
16	37.1	9.24	33.9	10.11
PC-cluster/TCGMSG				
1	230.6	1.00	230.6	1.00
2	121.6	1.90	119.4	1.93
4	64.9	3.55	61.7	3.74
8	35.0	6.59	32.8	7.03
16	25.1	9.19	18.9	12.20
PC-cluster/MPI-LAM				
1	...	...	272.6	1.00
2	...	...	140.3	1.94
4	...	...	72.6	3.75
8	...	...	38.6	7.06
16	...	...	21.9	12.45
PC-cluster[c]/TCGMSG				
1	...	...	228.5	1.00
2	...	...	127.0	1.79
4	...	...	66.6	3.43
8	...	...	37.0	6.17
16	...	...	22.6	10.11

[a] 旧並列化アルゴリズム
[b] 新並列化アルゴリズム
[c] Celeron 433MHz

```
 CALL MPI_ALLREDUCE(ARRAY,ARRAY1,LENGTH,MPI_DOUBLE_PRECISION,
1 MPI_SUM,MPI_COMM_WORLD,IERR)
 DO I=1,LENGTH
 ARRAY(I)=ARRAY1(I)
 ENDDO
```

のように、一旦仮の配列 ARRAY1 に結果が返され、仕様により ARRAY1 を ARRAY にすることは出来ず、ARRAY に戻すオーバーヘッドが生じているためである。今回はプログラミングの簡単化のためにこのような手順を採用したが、本格的に MPI に移行する場合には、このようなオーバーヘッドが生じないようにプログラムを書き直す必要があると思われる。

Figure 3 に並列度 16 の場合の各ノードでの CPU 占有時間を示した。これから容易にわかるように、旧アルゴリズムでは並列度が上がるにつれて、各ノードへの計算の効率的な分散がうまくいかなくなっており、特にノード 12、14 はほとんど使用されていない。一方、新アルゴリズムでは 16 ノードがほぼ均等に使用されており、負荷分散がうまく機能していることがわかる。

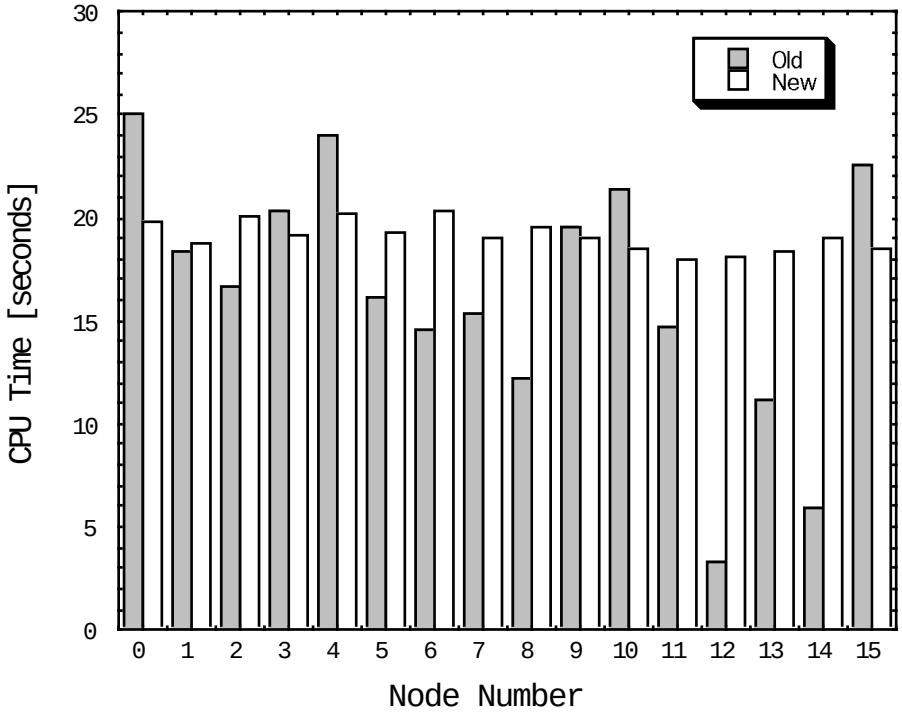

Figure 3. 16 ノード並列計算時の各 CPU 占有時間

CPU として Celeron 433MHz を用いると、並列化を行わない場合には Pentium II 450MHz とほぼ同等の結果が得られているが、並列化を行った場合には二割程度遅い結果となる。ここには示していないが他のベンチマークを走らせた場合にも同様の結果が得られている。Pentium II の 512KB と比べて二次キャッシュの容量が 128KB と小さいことと外部バスのスピードが遅い事が影響しているものと思われるが詳細は不明である。並列化を行った際に遅くなるため Celeron を PC クラスターに採用するのは好ましくないと考えられる。

以上の議論は前報 [4] を含めて CPU の占有時間のみの考察であったが、より重要なのは CPU の占有時間ではなく、実際の計算にかかった実時間であろう。より大きなポリマーである、poly-

Table 2. Poly-(para-phenylene sulfide) の並列計算の速度比較（秒）

プロセス数	CPU 時間 (秒)	CPU 加速率	実時間	実時間加速率
1	11640	1.00	13738	1.00
2[a]	6027	1.93	7899	1.74
2[b]	6016	1.93	12857	1.07

[a] 同一ハブに接続
[b] 別サブネット

(para-phenylene sulfide) を用いて実際の計算にかかった時間を比較した。Table 2 に RS6000/590 のクラスターを用いて計測した実時間について示した。実時間に関しては、ネットワークの構造や各 CPU の接続形態など、また計算時のネットワークの混雑の程度など、多くの因子が関係してくる。同一のハブに 2 台の WS を収容した場合には、実時間が 1.74 倍に加速されているが、別のサブネットにある 2 台で実行した場合には、途中のネットワークの状況などに大きく依存するようになり、ほとんど実時間の減少に結び付いていない事がわかる。従って例えば夜間に空いた PC を利用して計算を行おうというようなアイデアは上手くいかない可能性が高く、PC クラスターを用いた計算には同一のハブに接続するなどの専用のクラスターシステムをデザインする必要があると思われる。また同一のハブであっても上の結果のように 2 倍のパフォーマンスが得られていないことから、ネットワーク入出力待ちがかなりの程度生じていることが予想され、10BaseT ではなく 100BaseT でスイッチを使用する事が望ましい。現在、16CPU 程度の専用システムの構築を行っており、近い将来に詳細について発表する予定である。

参考文献

[1] M. Kertesz, *Adv. Quantum Chem.*, **15**, 161 (1982).

[2] P. Otto, E. Clementi, and J. Ladik, *J. Chem. Phys.*, **78**, 4547 (1983).

[3] H. Teramae, *J. Chem. Phys.*, **85**, 990 (1986).

[4] H. Teramae, *J. Chem. Software*, **4**, 73 (1998).

[5] R. J. Harrison, TCGMSG ver. 4.04, Battelle Pacific NorthwestLaboratory (1994)
R. J. Harrison, *Int. J. Quantum Chem.*, **40**, 847 (1991).

[6] Gregory D. Burns, Raja B. Daoud, James R. Vaigl,Supercomputing Symposium ’94 (Toronto, Canada, June 1994)
Greg Burns, Raja Daoud, MPI Developers Conference, University of Notre Dame (June 1995)

[7] P. J. Davis and P. Rabinowitz, *Method of Numerical Integration*, Academic Press, New York (1975), p.45.

[8] W. J. Hehre, R. Ditchfield, R. F. Stewart, and J. A. Pople, *J. Chem. Phys*, **52**, 2769 (1970).

[9] J. S. Binkley, J. A. Pople, W. J. Hehre, *J. Am. Chem. Soc.*, **102**, 939 (1980).
M. S. Gordon, J. S. Binkley, J. A. Pople, W. J. Pietro, W. J. Hehre, *J. Am. Chem. Soc.*, **106**, 2797 (1984).

[10] H. Teramae, *Theoret. Chim. Acta*, **94**, 311 (1996).

[11] 日向寺祥子、長嶋雲兵、青柳睦、佐藤三久、関口智嗣、桐山博史、細矢治夫、IPSJ SIG Notes, Vol. 94, 94-HPC-52, 19 (1994)

Parallel Processing on Ab Initio Crystal Orbital Calculations of One-Dimensional Polymers, Part 2

Hiroyuki TERAMAE

ATR Adaptive Communications Research Laboratories
2-2 Hikaridai, Seika-cho Soraku-gun Kyoto 619-0288, Japan
e-mail: teramae@acr.atr.co.jp

We have performed parallel processing on the ab initio crystal orbital calculations of one-dimensional polymers using the TCGMSG and MPI/LAM message passenger. An IBM RS6000 cluster (4 CPU), and personal computer cluster with Intel Pentium II 450MHz (2CPU) and Celeron 433MHz (2CPU) were used for the computational environment. We found that the Celeron CPU was less effective than Pentium II in the parallel calculations. Our previous work showed that the distribution of the CPU demand did not work well when 16 nodes were used to calculate the two-electron integrals in parallel. We found that the new algorithm incorporated in the present article improved the distribution. The CPU time increased when using the MPI interface, because there is an overhead to transfer the array in MPI_ALLREDUCE function. We also determined that the real through-put time was strongly dependent on the structure of the network system.

Keywords: Parallel processing, Crystal orbital, Energy band, PC cluster