

## パソコンクラスタによる分子の状態密度の並列計算

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  
お茶の水女子大学 理学部 化学科, 情報科学科, \*分子科学研究所 計算機センター,  
\*\*電子技術総合研究所 計算機方式研究室, \*\*\* (株)日立情報システムズ  
E-mail: {sachiko, umpei, hosoya}@is.ocha.ac.jp,  
aoyagi@draco.ims.ac.jp, {msato, sekiguchi}@etl.go.jp, kiri@crux.ims.ac.jp

### abstract

現在、大規模計算の必要性が強く求められているが、そのような計算を行うための高い計算コストは重大な問題である。本研究では計算コストの低下の可能性をみるため、安価なパーソナルコンピュータ (6台の日立 FLORA 3020H) からなるパソコンクラスタ上で、TCGMSG と PVM の2種類の異なったメッセージパッシングライブラリを用いて、巨大分子の状態密度計算の並列分散処理を試みた。その結果、スーパーニアスピードアップまたは、それに準ずる性能が得られ、高速計算機に並ぶ性能を引き出すことができた。また、パソコンクラスタ上で TCGMSG と PVM で問題のサイズによる違いは見られなかったが、ワークステーションクラスタ (6台のToshiba SUN IPC AS4040) では、問題のサイズが大きい場合、TCGMSG の性能が1.5倍程度悪くなった。

## Parallel Calculation of DOS for Large Molecule on PC Cluster

Sachiko HYUGAJI, Umpei NAGASHIMA, Mutsumi AOYAGI\*,  
Mitsuhisa SATOH\*\*, Satoshi SEKIGUCHI\*\*, Hirofumi KIRIYAMA\*\*\* and Haruo HOSOYA  
Ochanomizu University, \*Institute for Molecular Science,  
\*\*Electrotechnical Laboratory, \*\*\*Hitachi Information Systems  
E-mail: {sachiko, umpei, hosoya}@is.ocha.ac.jp,  
aoyagi@draco.ims.ac.jp, {msato, sekiguchi}@etl.go.jp, kiri@crux.ims.ac.jp

### abstract

In order to evaluate the cost-performance of large scale scientific calculation with a personal computer cluster system (PC cluster), the 6 HITAC FLORA 3020Hs system connected by Ethernet was tested. The density of states of large networks was calculated using two message passing libraries, TCGMSG and PVM. The performance of the program was also evaluated on a workstation cluster consisting of Toshiba SUN IPC AS4040 and a few supercomputers.

The PC cluster shows very good cost-performance compared with a workstation cluster and supercomputers. Two programs using TCGMSG and PVM show almost the same performance on PC cluster, whereas PVM shows better performance for large scale problem on the workstation cluster.

## 1. はじめに

ここ数年のワークステーションの性能向上の結果、従来以上に巨大計算が可能となってきた。さらに高速ネットワークで結合された複数台のワークステーション（ワークステーションクラス）上において並列分散処理をおこなうことによって、1台の計算機で実行した時の性能から期待される性能より高い性能（スーパーニアスピードアップ）を得ることができるようになり、コストパフォーマンスの面でも注目されている。さらに、問題によってはスーパーコンピュータに勝るとも劣らない性能を示すと報告されている<sup>1)</sup>。

一方、一般の企業や特に大学の研究室などでは、自由に一人が複数台のワークステーションを用いる環境をつくることは困難な状況にある。実際に、物理や化学で実験を中心に行っているところでも膨大な計算を必要としているが、限られた研究費のもと、小さな計算機資源を利用して莫大な時間を費やして精度の高い計算をしなければならない。これでは、数多くの大規模計算の要求に答えることができない。この解決方法の一つとしてさらに計算コストを下げる試み、つまり安価なパーソナルコンピュータによる並列分散処理が考えられる。ワークステーションレベルの計算機を複数台持つことが不可能であっても、既にパーソナルコンピュータを複数台所有している環境は数多くある。そしてその多くは、特に夜間などは有効に利用されていない。そのような既に所有している複数台のパーソナルコンピュータを用いて分散処理を行うことにより計算コストの削減が可能になると考えられる。

既にパーソナルコンピュータをサポートしている UNIX が数種存在するため、本研究では、その UNIX 上にメッセージパッシングライブラリをのせ、テスト的にネットワークにつながれたパーソナルコンピュータ、つまりパソコンクラスを構築し、分散処理を行った。また、パソコンクラスタの性能をスーパーコンピュータやワークステーションなどの高性能計算機の性能と比較するため、巨大分子の状態密度を計算するプログラム<sup>1223)</sup>を、パソコンクラスタやワークステーションクラスタ、スーパーコンピュータ上にて実装した。

さらに、メッセージパッシングライブラリの比較として、Toshiba SUN IPC AS4040 上ですでに実装している TCGMSG と、普及率の高い PVM を用いて計算を行った。

## 2. 実装条件

実装条件は表 1 及び表 2 に示すとおりである。

### 2.1. ハードウェア

用いた計算機は、パソコンクラスタ及びワークステーションクラスタである。参考のために、いくつかのスーパーコンピュータ上で、分散処理を行わないプログラムでその性能を測定した。

<表 1>

<Personal Computer>	日立 FLORA 3020H 6台	分子科学研究所計算機センター CPU : i486DX 33MHz Memory : 8MB×5台, 12MB×1台 Linpack値 : 1.2MFlops (100×100)
	NEC PC9821Ne 4台	分子科学研究所計算機センター CPU : i486DX 50MHz Memory : 14MB Linpack値 : 1.7MFlops (100×100)
<Workstation>	NEC EWS4800 330/310 4台/4台	分子科学研究所計算機センター Linpack値 : 10MFlops (330), 4.5MFlops (310) (100×100) *性能が異なるため 330 には 2 プロセスずつ生成
	Toshiba SUN IPC AS4040 12台	お茶の水女子大学教育用システム Linpack値 : 1.8MFlops (公称値 : 100×100)
<Supercomputer>	Cray C90	工業技術院情報計算センター
	NEC SX-3R	分子科学研究所計算機センター Linpack値 : 15MFlops (100×100), 130MFlops (1000×1000)

## 2.2. ソフトウェア

ソフトウェアは、TCGMSG4.0 と PVM3.2 を用いたが、NEC の PC9821Ne 上では OS に用いた 386BSD が XDR をサポートしていないため、PVM3.2 を実装することは不可能であった。そのため、PC9821Ne では TCGMSG のみを用いて性能を測定した。

<表 2>

機種	OS	メッセージパッシングライブラリ
日立 FLORA 3020H	LINUX 1.0.8	TCGMSG4.0, PVM3.2
NEC PC9821Ne	386BSD(98)	TCGMSG4.0
NEC EWS4800	SYSTEM V Rel.4.2	TCGMSG4.0, PVM3.2
Toshiba SUN IPC AS4040	SunOS 4.1.3	TCGMSG4.0, PVM3.2

## 2.3. プログラムの並列化

メッセージパッシングライブラリを用いたプログラムの概要を示す<sup>1)23)</sup>。このプログラムでプロセス相互に通信されるデータは  $NW \times 2$  ワードである。パソコンクラスタ及びワークステーションクラスタの性能測定の際は 1 台でも同じプログラムを用いたが、スーパーコンピュータ上では図中の \* の行のみの逐次処理用プログラムをコンパイラによる自動ベクトル化を用いて実行した。

<TCGMSG>

initialization (N,NW,Wmin,Wmax)*	
make MATRIX*	
call pbeginf	…並列環境の設定
call evon	
do i=1,N*	
if (mod(i,Nproc) .eq. Me) then	
calculate DOS*	…DOS の計算 (初期値問題の計算)
endif	
enddo*	
call dgop (RESULTS)	…結果の収集
call pend	
call fexit	…並列環境の終了

<PVM>

マスター側		スレーブ側
initialization (N,NW,Wmin,Wmax)*		
call pvmfspawn	→	
make MATRIX*		make MATRIX
do i=1,N*		do i=1,N
if (mod(i,Nproc) .eq. 0) then		if (mod(i,Nproc) .eq. Me) then
calculate DOS*		calculate DOS
endif		endif
enddo*		end do
		call pvmfinitend
		call pvmfpack (RESULT)
	←	call pvmfsend (master,RESULT)
do i=1,Nproc-1		
call pvmfrecv (RESULT)		
call pvmfunpack		
enddo		
sort RESULT		
call pvmfexit		call pvmfexit

### 3. 計算結果

紙面の都合上、日立 FLORA によるクラスタ上での結果を中心に述べる。

#### 3.1. パソコンクラスタ上での TCGMSG と PVM による並列化の効果

パソコンクラスタとワークステーションクラスタ上では、データのサイズ ( $N \times N$ ) を  $22 \times 22$  から  $20482 \times 20482$  まで、ループの繰り返し回数及び通信データ量 ( $NW$ ) を 10 から 320 まで変化させてその時間を計測した。各プロセスで多少時間の差が見られたため、その各プロセスの CPU 時間の平均を計測した。また、スーパーコンピュータ上では、 $N$  と  $NW$  を 6 点のみ (表 3) に固定し、1 プロセスにて実行させた CPU 時間を測定した。

6 台の日立 FLORA により構成されたパソコンクラスタ上で、巨大分子の状態密度を計算するプログラムを実装した結果のうち、データサイズの大きいものと小さいものを下に示す。横軸に生成したプロセスの数、縦軸に CPU 時間の逆数 (スピードアップ) をプロットし、1 台で計算した性能から期待される直線的に向上する性能 (リニアスピードアップ) を直線で表わしたグラフを示す。ここで、1 プロセッサでは 1 プロセスのみが実行するようにした。すなわち、プロセッサ数=プロセス数となる。

図 1  $N=22, NW=40$  の時の実行時間

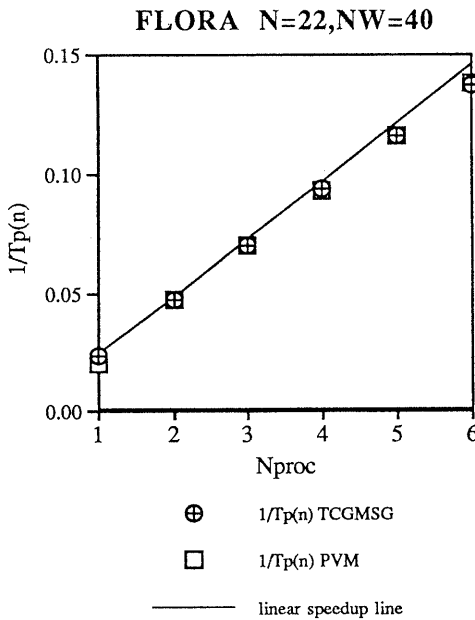
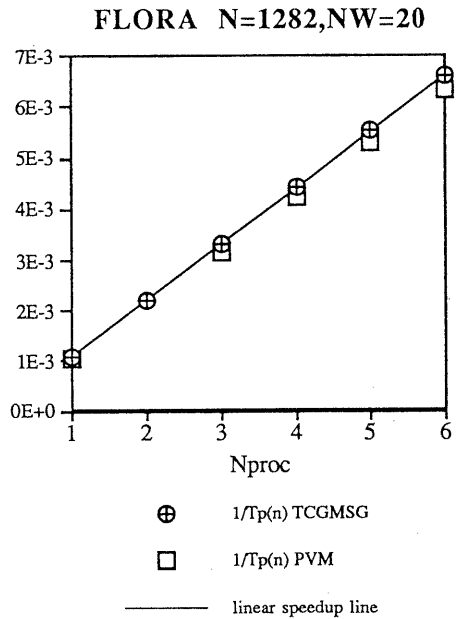


図 2  $N=1282, NW=20$  の時の実行時間



PVM で実装した場合には、1 台で計算した結果から期待される性能よりも高い性能 (スーパーリニアスピードアップ) が得られた。TCGMSG 上で実装した場合には、ほぼリニアスピードアップが観測されたが、小さなデータサイズの問題の計算においてはデータサイズが大きい場合と比較すると性能向上の度合いがわずかに悪くなる。大まかに見ると、TCGMSG と PVM で大きな差は見られなかった。これは、用いたプログラムが、データ転送にかかる通信量が少ないプログラムであったためであろう。もう少し細かに実測された CPU 時間で比較をすると、1 台で実行したときには TCGMSG の方がよい性能を示すが、台数を増やすにつれ、PVM の性能が向上している。

双方のライブラリのインターフェースに多少の差があったため両者のプログラミングに若干の違いが生じた。そのため、TCGMSG と PVM の差はライブラリによる差であるのか、プログラミングによる差であるのかは今後解析する。

### 3.2. 日立 FLORA と他機種との比較

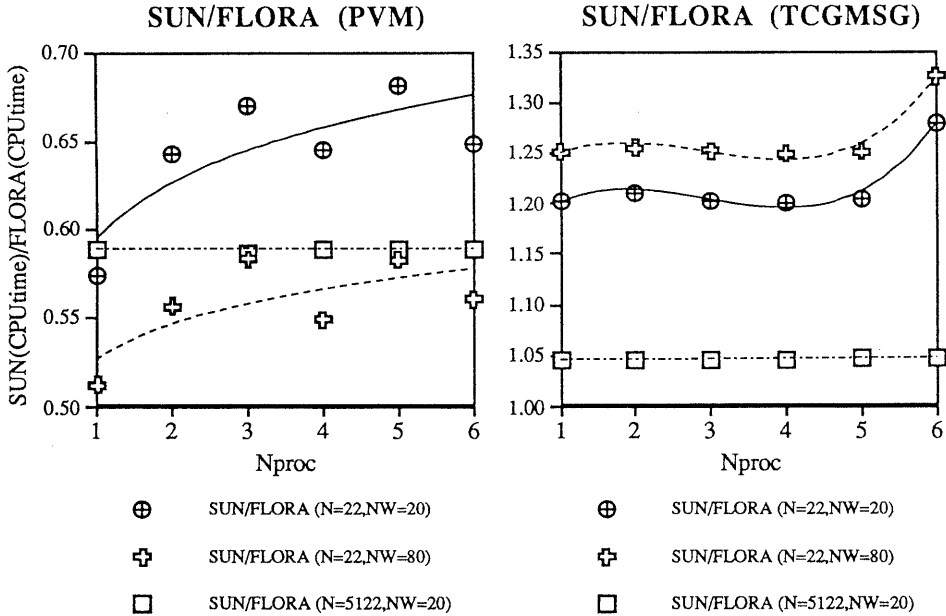
本研究では6台のパーソナルコンピュータによるパソコンクラスタ上でテスト的にメッセージパッシングライブラリを用いて実行させた。台数を増やしたときにはオーバーヘッドが増加することが考えられるが、6台によるテスト段階ではオーバーヘッドは無視できる範囲であったため、この章での他機種との比較はパソコンクラスタの性能が直線的にのびるものと仮定して行った。

#### <日立 FLORA vs Toshiba SUN IPC AS4040>

次に示すのは横軸にプロセス数、縦軸にToshiba SUN IPC AS4040 上での CPU 時間を日立 FLORA 上での CPU 時間で割ったものをプロットした図である。グラフ上の線は点をフィッティングしたものである。このグラフでは点が高いところになるほど、FLORA の性能が高いことを示している。

図3 PVM を用いたときの問題の違いによる比較

図4 TCGMSG を用いたときの問題の違いによる比較



PVM を用いて実装した場合には、データサイズが小さな際の台数効果は台数が少ないところで急な立ち上がりを見せすぐに飽和しているが、データサイズが大きな際は台数が増えても SUN と FLORA の比はほとんど変化がない。この問題サイズによる違いは、パーソナルコンピュータのメモリアーキテクチャがワークステーションと比べて大きな問題向けに設計されていないために起こるものと考えられる。問題サイズが小さいときには、ワークステーションでは1台目からメモリアーキテクチャ上のネックは現われないが、パーソナルコンピュータでは2台目近辺でメモリアーキテクチャ上のネックが解消され良いスピードアップが観測される。ところが、大きな問題を解く際には台数を増やしてもそのネックがなかなか解消されないために、グラフの立ち上がりが大きな台数のところになると考えられる。

TCGMSG を用いて実装した場合には、SUN の性能が大幅に落ちるため、FLORA の性能が SUN の性能を越えてしまう。この現象は問題サイズを変化させても顕著に見られるが、この SUN 上の TCGMSG の性能についての詳しい解析はまだ行っていない。

#### <日立 FLORA vs NEC SX-3R>

分子科学研究所計算機センターのスーパーコンピュータ NEC SX-3R を用いて同じプログラム (2.3. プログラムの並列化参照) を計算し、その CPU 時間を測定した。SX-3R 上で実装した結果と日立 FLORA 6 台によるパソコンクラスタ上で実装した結果を下の表3に示す。()内は SX-3R に対して日立 FLORA で同じ性能をだすためには、何台から構成されるパソコンクラスタが必要かを示した数値である。ここで、用いたプログラムの通信量が少ないために直線的に性能が向上することが可能となるため、日立 FLORA は6台によって構成されるパソコンクラスタ上での CPU 時間を基準として、直線的に性能が向上するものと仮定して計算している。

&lt;表3&gt;

問題サイズ	NEC SX-3R (sec)	日立 FLORA 6台	
		TCGMSG (sec) (台)	PVM (sec) (台)
N=22, NW=40	0.7	7.28 (62)	7.20 (62)
N=22, NW=80	1.5	18.19 (73)	18.16 (73)
N=22, NW=160	3.4	50.71 (89)	51.00 (96)
N=1282, NW=20	9.9	151.05 (92)	157.62 (96)
N=2562, NW=20	21.3	299.70 (84)	321.52 (91)
N=5122, NW=20	39.8	566.35 (85)	599.20 (90)

表1に示したようにLinpackの値では、小さな問題ではパーソナルコンピュータとスーパーコンピュータの性能の比は約10であるが、大きな問題の時はパーソナルコンピュータの性能が小さい問題の時と変わらないと仮定すると約100となる。表3から分かるように、問題サイズが大きくなるに従って、スーパーコンピュータの性能と並ぶために必要となるパーソナルコンピュータ台数が増加している。これは、Linpackの値から想像されるように、パーソナルコンピュータのアーキテクチャは大きな問題を解くように作られておらず、一方でスーパーコンピュータは大きな問題を解くときにその性能が十分発揮されるようにアーキテクチャが設計されていることを示唆している。

#### 4. まとめ

本研究では、2種類のメッセージパッシングライブラリ(TCGMSG, PVM)を用いて、パーソナルコンピュータ6台よりなるパソコンクラス上で、巨大分子の状態密度を求めるプログラムの並列分散処理を行った。その結果、メッセージパッシングライブラリによる大きな差は見られなかったが、問題サイズにより、それぞれのライブラリでわずかな差が見られた。二つの異なるライブラリを用いることにより生じる差は、ライブラリの性能による差であるのか、プログラミングによる差であるのかは、今後解析する必要がある。Toshiba SUN IPC AS4040上でTCGMSGを用いて実行した場合には他の場合と比べて性能がかなり落ちるため、日立FLORAの性能の方が高く現われる。

パーソナルコンピュータのメモリアーキテクチャはワークステーションに比べて大きな問題を解く目的で設計されていないため、問題サイズを大きくしたときの計算機の性能に対する負荷がワークステーションに比べて大きく、性能がでにくくなっている。その傾向は、スーパーコンピュータと比較したときにはさらに大きく現われる。

#### 5. 今後の課題

今回の研究では、小規模のパソコンクラスを用いてテスト的に並列分散処理を行うにとどまったが、今後は大きなパソコンクラスを用いて並列分散処理を行い実用化の可能性を見る。また、台数増加によるオーバーヘッドに対する考察も深める必要がある。さらに、TCGMSGとPVM双方のメッセージパッシングライブラリの詳細な比較も行いたい。

化学的な見地からは、現在計算しているベンゼンが1次元的につながったポリアセン系分子から、2次元に広がる巨大なグラファイト系分子に対象を拡大して状態密度の計算を実行する。

#### 6. 謝辞

本研究を行うにあたり、分子科学研究所計算機センター、工業技術院情報計算センターに感謝いたします。また、分子科学研究所計算機センターの高見利也博士、東京エレクトロンの雪竹潤氏にも心より感謝いたします。

#### 7. 参考文献

- 1) 日向寺祥子, 長嶋雲兵, 細辻治夫, 関口智嗣, 佐藤三久, IPSJ SIG Notes, Vol.93, 93-HPC-48, 8 (1993)
- 2) M. L. Williams and H. J. Maris, *Phys. Rev. B*, 31, 4508 (1985)
- 3) K. Yakubo and T. Nakayama, *Phys. Rev. B*, 36, 8933 (1987)
- 4) 関口智嗣, 長嶋雲兵, 日向寺祥子, IPSJ SIG Notes, Vol.93, 93-HPC-47, 13 (1993)