

# 地球シミュレータによる蛋白質の大規模シミュレーション： ベクトル化と並列化による加速性能

齋 藤 稔<sup>†</sup>

非常に巨大な蛋白質（水中のヘモグロビン：全原子数 12 万 36 個）について、分子動力学（MD）シミュレーションのスピードを地球シミュレータと COSMOS90 を用いて測定した。COSMOS90 は、コード全体をベクトル化および並列化した。COSMOS90 内部の Barnes-Hut tree code は、ベクトル化を維持したまま並列化した。その結果、128 台のプロセッサで 0.023 sec/step のスピードを達成した。このスピードならば、生命科学で重要なヘモグロビンの機能をシミュレーションで研究することが可能である。さらなる高速化のために、COSMOS90 のサブプロセスごとに、ベクトル化と並列化による加速性能を調査した。

## Large Scale Simulations of Proteins on the Earth Simulator: Acceleration Performance by Vectorization and Parallelization

MINORU SAITO<sup>†</sup>

I measured a performance speed of large scale molecular dynamics (MD) simulations for a protein (hemoglobin in water; 120,036 atoms) using COSMOS90 on the Earth Simulator. The code of COSMOS90 was highly vectorized and parallelized on the Earth Simulator. The Barnes-Hut tree code in COSMOS90 was parallelized in keeping with the vectorization. I successfully obtained the performance speed of 0.023 sec/step for 128 processors of the Earth Simulator. This performance speed makes it possible to study biological properties of hemoglobin by MD simulations. For further speed up, I investigated acceleration performances by vectorization and parallelization for each subprocess of COSMOS90.

### 1. はじめに

地球シミュレータには、現在、様々なアプリケーションプログラムが移植され実行されているが、アプリケーションプログラムによる性能評価の報告は非常に少ない。これまでの性能評価は、主に地球シミュレータセンター内部のアプリケーションプログラムに限られていた。性能評価といえども地球シミュレータセンターに出向き、性能評価用のたくさんのジョブを投入しなければならない。アプリケーションプログラムによる地球シミュレータの性能評価は、他の分野のアプリケーションプログラムの開発者やユーザが、プログラムを移植し高速化するとき、あるいは次世代のスーパーコンピュータの開発計画を推進するとき有用な資料になる。筆者は、バイオ分野のアプリケーションプログラムの開発者およびユーザとして、今後、急激に増大すると予想されるバイオ分野の大規模計算について、地

球シミュレータの性能評価を行った。

蛋白質の分子動力学（MD）シミュレーションは、1977 年に世界で初めて行われた<sup>1)</sup>。それ以来、スーパーコンピュータの性能向上とともに普及してきた。現在では、蛋白質の標準的な計算手法として、バイオ分野の研究者に広く用いられている。蛋白質の MD シミュレーションは、多体問題をコンピュータで扱う粒子シミュレーションの一種である。蛋白質を 1 個 1 個の原子の集まりとしてとらえる。それによって、蛋白質の物性や機能を原子の運動から理解し予測するのが目的である。蛋白質のシミュレーションは、固体や液体などの他の多体問題のシミュレーションよりも複雑で計算量も膨大である。その理由は、第 1 に、蛋白質は原子が連結した高分子であるために、原子間に複雑な原子間力（2 体力（bond）、3 体力（angle）、4 体力（torsion））が働いている。第 2 に、蛋白質は周囲の環境（水溶液や生体膜）に敏感であるために、環境を含めてシミュレーションしなければならない。蛋白質は、これらの複雑な原子間相互作用と環境との相互作用とがバランスしながら、固有の立体構造を持ち機能して

<sup>†</sup> 弘前大学理工学部

Faculty of Science and Technology, Hirosaki University

いる。したがって、蛋白質のリアルなシミュレーションは、複雑な相互作用を持つ不均一で巨大な系を対象にしなければならない。これらの膨大な計算の一部を無視したり、安易に単純化すると、計算量を低減させることができるが、これらの相互作用のバランスを崩すので、シミュレーションの信頼性は目覚ましく損なわれる。その結果、蛋白質をリアルにシミュレーションすることは、できなくなる。

蛋白質は環境に敏感で複雑な高分子であるために、その現象は、一様な固体や液体に比べて非常に広い時間・空間・エネルギーの領域に及んでいる。空間では1 Åから数百Å、時間では1 フェムト秒から1 ミリ秒、エネルギーでは、1 kcal/mol から1 eVに及ぶ。スーパーコンピュータの登場によって、蛋白質を高速にシミュレーションすることが可能になり、広い時間・空間・エネルギー領域の蛋白質の現象を解析・予測することが可能になった。実際に、1977年に始まった蛋白質のMDシミュレーションは、1980年代のベクトル型スーパーコンピュータの登場によって、大きく進歩した。すなわち、それまで真空中で行われていたシミュレーションを、水溶液中で行うことができるようになった。

現在の最速のスーパーコンピュータは、地球シミュレータである(2004年9月まで)。640個のノードからなり、各ノードは8個のベクトルプロセッサからなる。各プロセッサのピーク演算性能は8 Gflopsであり、全体のピーク性能は40 Tflopsである。ノード内の8プロセッサは、それぞれ32 GB/secのバンド幅でメモリを共有しており、640個のノードは、単段クロスバスイッチで結合しており、理論最大スループットは12.3 GB/secである。

この研究の目的は、地球シミュレータを利用することによって、どの程度の規模の蛋白質のシミュレーションが可能になるかを調べることである。通常の規模の蛋白質(水溶液中のDNA蛋白質複合体:全原子数約16,034個)を共同利用のスーパーコンピュータ(計算科学研究センターのVPP5000, 8プロセッサ)で実行した結果と比較した<sup>2)</sup>。その結果、地球シミュレータの64プロセッサで、約4倍のスピードを達成した。今回の研究では、これまで、だれもなしえなかった規模の蛋白質(水中のヘモグロビン:図1)について、地球シミュレータにおけるベクトル化と並列化による加速性能を評価した。

## 2. 蛋白質のシミュレーションのプログラム

蛋白質のシミュレーションのプログラムは、すでに数種類が世界で普及している。米国のCHARMM、

AMBER, NAMD, 欧州のGROMOSなどである。これらのうち、AMBERは、日本国内で早い時期に普及しており、並列化の検討も国内で早くからなされていた<sup>3)</sup>。日本国内では、国内産の2つのプログラム(COSMOS90, PRESTO)もまた、よく利用されている。COSMOS90は、1990年に開発された<sup>4)</sup>。当時のベクトル型スーパーコンピュータ(fujitsu VP400EやVP2600)のベクトルプロセッサの能力を引き出して、高速にシミュレーションするために開発された。蛋白質のシミュレーションで最も時間のかかる長距離クーロン力を $O(N \log N)$ で高速に計算することを可能にした世界で最初のプログラムである。これによって、水溶液中の蛋白質を長距離クーロン力をカットオフ(cutoff)せずに、長時間シミュレートすることが初めて可能になった。その後、COSMOS90は、ベクトル型並列スーパーコンピュータ(fujitsu VPP500, VPP5000)へVPP-Fortranによって並列化され移植された<sup>5)</sup>。

海外のプログラム(CHARMM, AMBER, NAMD, GROMOS)は、すべてスカラ型並列計算機のために開発されているので、ベクトル型並列スーパーコンピュータでの高速化の研究はほとんど行われていない。したがって、海外のプログラムは、地球シミュレータへの移植をして、高速化する目的には向いていない<sup>5)</sup>。ただし、AMBERは日本国内にユーザが多いために、国内のベンダ(fujitsu, NEC)が自社のベクトル型計算機に移植している。しかし、まだ、ベクトル型高並列計算機の能力を十分に引き出すにはいたっていない。COSMOS90とほぼ同時期に国内で開発されたもう1つのプログラム(PRESTO)も、ベクトル化されたプログラムであるが、蛋白質の大規模シミュレーションよりは、むしろ他のプログラムと組み合わせでモデリングを行うために開発されたプログラムである。そのほかにも、近年、国内産プログラム(PEACHなど)が開発されたが、そのうちのいくつかはベクトル化も試みられているものの、いずれもスカラ型並列計算機向けに開発されたプログラムである。したがって、地球シミュレータのベクトル型並列計算機としての性能を引き出すためには、COSMOS90を高速化するのが最も適している。

## 3. プログラムの構造

蛋白質の分子動力学シミュレーションのプログラムはいくつかあるが、共通した構造を持っている。まず、時間を進めるためのループがある。ループの中では、原子に働く力を計算し、運動方程式に従って原子の座標を更新する。時間ループは、微小な時間刻み

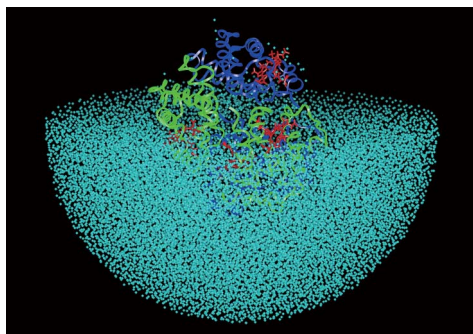


図 1 水中のヘモグロビン (120,036 原子). 水球の半径は 66 Å  
Fig. 1 Hemoglobin in water (120,036 atoms). Radius of water sphere is 66 Å.

### Loop for MD step

#### (1) Calculating bonded forces (bond, angle, torsion)

#### (2) Communicating bonded forces

#### (3) Calculating nonbonded forces based on PPPC

##### (3.1) Subdividing a system into hierarchical cubic cells

##### (3.2) Communicating child cells and next cells

##### (3.3) Making an interaction table for P-P and P-C

##### (3.4) Communicating dipoles

##### (3.5) Calculating charges and dipoles of cells

##### (3.6) Communicating charges and dipoles of cells

##### (3.7) Calculating Coulomb and vdW forces from P and C

#### (4) Updating positions according to eq. of motion

#### (5) Communicating new positions

### Next step

図 2 COSMOS90 は, PPPC 法に基づいて長距離クーロン力を高速に計算してシミュレーションを行う

Fig.2 Structure of COSMOS90. COSMOS90 simulates protein dynamics by efficiently calculating long-range Coulomb interactions using PPPC method.

$\Delta t \approx 10^{-15}$  sec を単位にして, シミュレーションの目的に応じて  $10^5$  から  $10^9$  回程度まわる. 力の計算は, 力の種類に対応したサブルーチンを次々に呼んで実行する. したがって, 一般に, 蛋白質のシミュレーションにかかる時間は, ループの回数と 1 回あたりのループで実行される力の計算に要する時間で決まる. 運動方程式を解く計算は, 力の計算に比べて時間はかからない. ループの回数は, シミュレーションする時間の長さに単純に比例する. また, 力の計算にかかる時間は, シミュレーションする蛋白質の大きさに依存する. したがって, 巨大な蛋白質を高速にシミュレーションするためには, ステップごとに行われる力の計算をベクトル化や並列化によって短縮しなければならない.

力には, 結合性の力と非結合性の力とがある. 結合

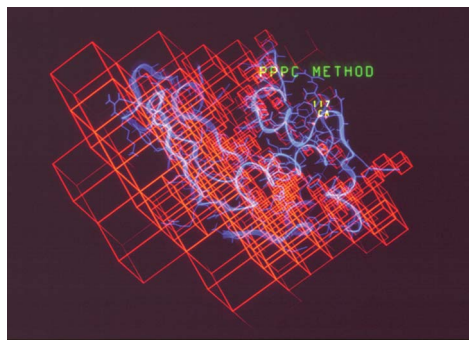


図 3 Particle-Particle and Particle-Cell (PPPC) 法によって長距離クーロン力を  $O(N \log N)$  で計算する. 図は, ヒトリゾチーム蛋白質 (130 アミノ酸: リボンで表示) の 117 番目の  $C\alpha$  炭素原子に加わるクーロン力を計算する際の空間分割を示す

Fig.3 Long-range Coulomb interactions are calculated by the order  $N \log N$  using Particle-Particle and Particle-Cell (PPPC) method. This figure shows space subdivision based on PPPC to calculate Coulomb forces acting to the 117th  $\alpha$ -carbon atom of human lysozyme (130 amino acids: displayed by ribbon).

性の力 (bond, angle, torsion) は, それぞれ複雑な 2 体, 3 体, 4 体力であるが, 短距離力であるために, 全体の計算量は蛋白質の全原子数  $N$  にほぼ比例する. 一方, 原子間に働く非結合性の力には, van der Waals 力とクーロン力がある. vdW 力は, 距離の 7 乗に逆比例し急激に減衰するために, 打ち切り (cutoff) の近似が有効である. 一方, クーロン力は, 距離の 2 乗に逆比例する長距離力であるために, 打ち切りするべきではない. したがって, 一般に, クーロン力は, 全原子が相互作用する 2 体力であるために, その計算量は  $N^2$  に比例して増大する. その結果, 巨大な蛋白質のシミュレーションで, 最も時間がかかるのは, クーロン力の計算である. 蛋白質のシミュレーションのプログラムは, 共通の構造を持っているが, クーロン力の計算方法がお互いに異なっている. たとえば, AMBER は Particle-Mesh-Ewald (PME) 法, NAMD は First-Cell-Multipole (FCM) 法, COSMOS90 は Particle-Particle and Particle-Cell (PPPC) 法を使っている. PME 法は, 周期的境界条件でのみ利用可能である. 一方, FCM 法と PPPC 法は, 周期的境界条件と有限境界条件 (自由境界条件を含む) の両方で利用可能である. ただし, COSMOS90 は, PPPC 法を周期的境界条件下で利用するようにしていない.

## 4. COSMOS90 の並列化

COSMOS90 は, 長距離クーロン相互作用を

$O(N \log N)$  で高速に処理し、カットオフせずにシミュレーションすることを可能にした最初のプログラムである。COSMOS90 の構造を図 2 に示す。COSMOS90 は、他のプログラムと同様に時間ループの中に、(1) 結合性力の計算 (3) 非結合性力の計算 (4) 運動方程式の計算からなっている。ただし、COSMOS90 では、原子と水分子を同一の粒子として取り扱う。水分子の重心を Cartesian 座標、方位を Quaternion で記述している。他のプログラムと COSMOS90 との大きな違いは、長距離クーロン力を PPPC 法で  $O(N \log N)$  で処理することである。PPPC 法では、図 2 のようにクーロン力を計算する。すなわち、粒子と近隣の粒子および粒子と遠方のセル（粒子を含む）との相互作用を計算する。PPPC 法では、まず、系を含む空間全体を Barnes-Hut tree code<sup>6)</sup> で入れ子になったセルに分割する（図 2 の (3.1)）。すべてのセルは、8 分木の node に対応する。次に、粒子（8 分木の葉）と相互作用する近隣の粒子や遠方のセルとの相互作用表を、Barnes-Hut tree に沿って探索することによって作成する（図 2 の (3.3)）。具体例を図 3 に示す。相互作用表に従って、粒子やセルの持つ電荷や双極子から粒子に働く力を計算する（図 2 の (3.5)）。

PPPC 法では、上記の空間分割以外にも原子や分子の運動の特性を利用して高速化を行っている<sup>4)</sup>。第 1 に、近隣の粒子からのクーロン力は激しく変化するが、遠方の粒子やセルからのクーロン力は緩やかに変化する。したがって、図 2 の (3.7) の遠方からのクーロン力の計算は、毎ステップ行う必要がない。第 2 に、蛋白質の構成原子と周囲の水分子は、ともに激しく振動しながら、ゆっくりと場所を移動する。したがって、Barnes-Hut 空間分割とそれに基づく相互作用表は、毎回更新する必要がない。COSMOS90 では、図 3 の系についての誤差解析から、図 2 の (3.1)(3.2)(3.3) を 50 ステップごとに、(3.4)(3.5)(3.6)(3.7) を 10 ステップごとに行っている<sup>4)</sup>。これらの頻度は、計算対象によって変えない。

図 2 で示す COSMOS90 のすべての処理について、ベクトル化と並列化を行った。並列化は、結合性の力 (bond, angle, torsion) の計算と 8 分木による空間分割の計算を除いて、粒子についてのループをサイクリックに分割してプロセッサに割り当てている。結合性の力 (bond, angle, torsion) は、それぞれのリストに対してバンド分割をして並列処理し、計算結果を allgather によって全通信をしている（図 2 の (2)）。

Barnes-Hut tree に基づく空間分割は、Makino のアルゴリズム<sup>7)</sup> によって高度にベクトル化されている。

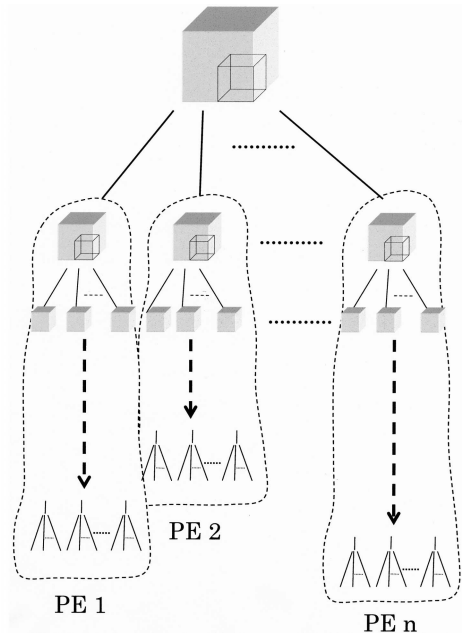


図 4 Barnes-Hut 空間分割の並列化

Fig. 4 Parallelization of Barnes-Hut space subdivision.

ベクトル化による高速化を犠牲にしないために、このアルゴリズムを保持したまま、図 4 に示すように並列化した。すなわち、下位レベルのセルを別々のプロセッサが担当して、分割をすすめる（図 2 の (3.1)）。各プロセッサが持つ木構造情報のうち子セルと次セルの情報のみを全通信する（図 2 の (3.2)）。総プロセッサが 8 台以上 64 台未満の場合には、ルートセルの 8 個の子セルを 8 個のプロセッサに割り振る。総プロセッサが 64 台以上の場合には、ルートセルの孫セル 64 個を、64 台のプロセッサに割り振る。8 分木の構築に参加しなかったプロセッサは、子セルと次セルの情報のみを受け取る。粒子と相互作用する近隣の粒子や遠方のセルをリストアップするために、その粒子を担当するプロセッサは、8 分木をルートセルから上位のレベルへ探索する。このときに、全通信によって得ている子セルと次セルの情報のみを利用する。その結果、相互作用表は、粒子を担当するプロセッサに分散して保持される。相互作用表はプログラムの中で最も大きな配列であるから、すべてのプロセッサに分散することによって、巨大な蛋白質をシミュレーションする際にメモリがボトルネックとなることを回避できる（図 2 の (3.3)）。

次に、Barnes-Hut tree を構築したプロセッサが、すべてのセルの電荷と双極子を 8 分木を下から上にたどることによって並列計算する（図 5）。得られた電荷と双極子をすべてのプロセッサに対して全通信する

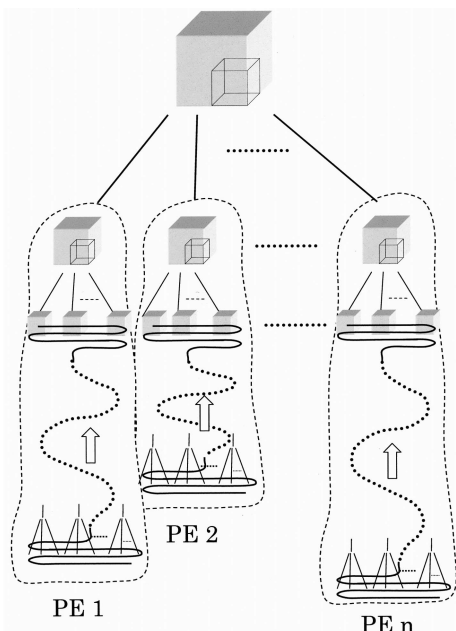


図 5 セルの電荷と双極子の並列計算

Fig. 5 Parallel calculation of charges and dipoles for cells.

(図 2 の (3.6)). 相互作用表を参照して、粒子と近傍の粒子、粒子と遠方のセルとの相互作用を並列計算する (図 2 の (3.7)). 次に、結合性の力と非結合性の力の合力に対して、運動方程式を解き、すべての粒子について新しい座標を求める (図 2 の (4)). 最後に、粒子の新しい座標を全通信することによって、すべてのプロセッサが、すべての粒子の新しい座標データを持つ。

### 5. ベクトル化と並列化による加速結果

並列化した COSMOS90 を地球シミュレータに移植して、ベクトル化と並列化による加速性能を調べた。地球シミュレータのノード内外のプロセッサの違いを区別せずに、MPI によって並列化した (MPI フラットプログラミング)。ベクトル化による加速性能を明らかにするために、同一のソースコードをベクトルコンパイルとスカラコンパイルした 2 種のロードモジュールを用意して、それぞれの実行時間を測定した。一方、並列化による加速性能を明らかにするために、プロセッサ台数を 1 台から 128 台 (16 ノード) に変えて、実行時間を測定した。実行時間の測定は、MPI\_Wtime によって行った。2 種の規模の蛋白質を計測に用いた。1 つは、DNA と蛋白質 ( $\lambda$ -repressor) とが結合した複合体が、34 Å の水球に浸っている。全原子は 16,034 個 (複合体原子数 2,225 個、水分子数 4,603 個) であり、現在の共同利用のスーパーコンピュー

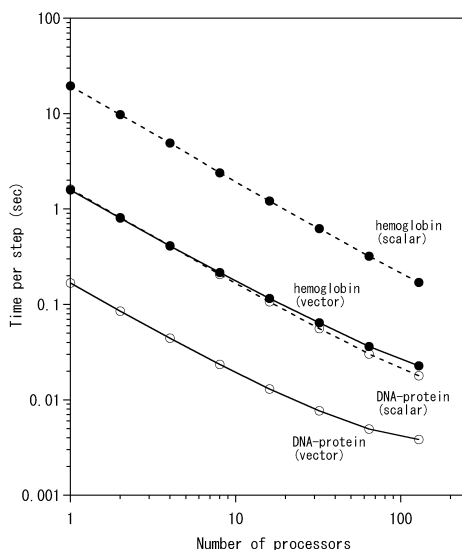


図 6 地球シミュレータ上の COSMOS90 の計算スピード  
Fig. 6 Calculation speed of COSMOS90 on the Earth Simulator.

タ (たとえば、VPP5000 の 8 プロセッサ) を用いて、実際のバイオの研究が可能である<sup>8)</sup>。もう 1 つは、半径 66 Å の水球に浸ったヘモグロビンである。全原子数が 12 万 36 原子 (蛋白質原子数 9,066 個、水分子数 36,990 個) であり、共同利用のスーパーコンピュータの能力では、十分なシミュレーションを行うことができない。これまで、バイオの分野では、簡略化したシミュレーションが行われていたが、リアルなシミュレーションは、まだ行われていなかった<sup>9)</sup>。

測定結果を図 6 に示す。スカラ計算を破線でベクトル計算を実線で示している。ヘモグロビンのベクトル計算は、DNA 蛋白質複体のスカラ計算の曲線とほぼ重なっている。ヘモグロビンと DNA 蛋白質複体との計算は、いずれも 128 プロセッサまで滑らかに加速している。ヘモグロビンのベクトル計算のスピードは、128 台のプロセッサで 0.023 sec/step であった。米国の NAMD2.4 ならば、5 分の 1 の規模の蛋白質 (2 万 3 千 558 原子) でのスピードである<sup>10)</sup>。COSMOS90 の計算速度ならば、数千万ステップのシミュレーションが、約 1 週間で実行することができる。したがって、実際のバイオの研究を行うことができる十分なスピードである。

たとえば、突然変異によるヘモグロビン機能の変化をシミュレーションで予測する。具体的には、ヘモグロビンのアミノ酸をシミュレーションで置換することによって、酸素吸着の親和力変化を実験に先立って計算する。また、酸素や一酸化炭素の吸着解離をシミュ

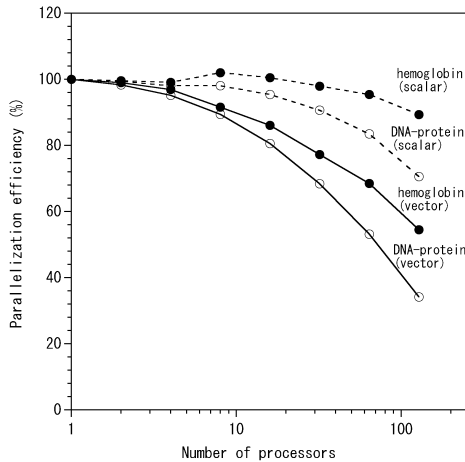


図7 並列化効率

Fig. 7 Parallelization efficiency.

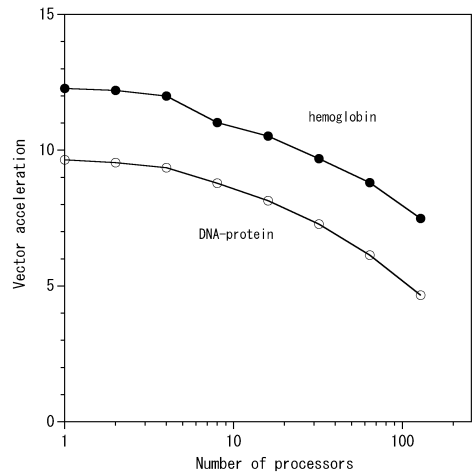


図8 ベクトル化による加速

Fig. 8 Acceleration by vectorization.

レーションで追跡し、構造変化の過程を明らかにする<sup>11)</sup>。

一方、DNA 蛋白質複合体 (16,034 原子) は、現在行われている研究と同程度の規模であるが、ベクトル計算のスピードは 128 プロセッサで 0.0035 sec/step であった。 $10^9$  ステップのシミュレーションを、40 日のできるスピードである。すなわち、これまで、困難であった中規模の蛋白質の構造形成がコンピュータで追跡できるスピードである。

DNA 蛋白質複合体の並列化による加速 (図 6) は、128 プロセッサ台数付近で減衰しているように見える。ベクトル化と並列化による加速性能を、それぞれ詳しく知るために、図 7 に、ベクトル計算とスカラ計算について、並列化効率 (すなわち、並列化加速 ÷ プロセッサ台数) をプロットした。ヘモグロビンと DNA 蛋白質複合体とは、ともにスカラ計算 (破線) が、ベクトル計算 (実線) よりも高い並列化効率を示すことが分かる。同時に、ヘモグロビンは、ベクトル計算とスカラ計算のいずれに対しても、DNA 蛋白質複合体よりも高い並列化効率を示す。並列化効率が図 7 に示すように計算規模とベクトル化に依存する理由は、通信に対する演算の粒度の割合が変わるためであると考えられる。

図 8 にベクトル化による加速性能を示した。ヘモグロビンについてのベクトル化加速性能は、DNA 蛋白質複合体よりもすべてのプロセッサ台数において優れている。しかし、同時に並列化効率と同様に、プロセッサ台数の増加とともに低下している。このように、ベクトル化加速が、プロセッサ台数や問題サイズに依存する理由は、ベクトル化によって加速されない通信処

理の割合が演算処理の割合に対して相対的に増大するためであると考えられる。しかし、同時に、プロセッサ台数の増加によってループの分割がより細分化され、その結果ループ長が減少してベクトル化加速が低下している可能性もある。

並列化とベクトル化による加速をさらに詳細に分析するために、図 2 のサブプロセスごとに、図 6 と図 8 と同様の計測を行った (図 9 と図 10)。ただし、図 9 と図 10 の communication とは、図 2 のすべての通信にかかる時間を合計したものである。また、communication には、mpi 通信関数を呼ぶ前後で行っているデータの収集分散のためのコピーの時間を含んでいる。最も時間がかかる演算は、第 1 に、非結合性の力を計算するサブプロセスである (nonbonded force, 図 2 の (3.7))。第 2 に、非結合性の力を計算するための相互作用表の作成 (nonbond table, 図 2 の (3.3))。第 3 に、セルの電荷と双極子の計算 (charge-dipole, 図 2 の (3.5))。第 4 に、結合性の力の計算 (bonded force, 図 2 の (1))。第 5 に、運動方程式の計算 (eq.of motion, 図 2 の (4))。最後に、Barnes-Hut 空間分割である (図 2 の (3.1))。すべての演算の処理時間は、運動方程式の計算と Barnes-Hut 空間分割を除いて、プロセッサ台数の増加とともに理想的に減少している。

Barnes-Hut 空間分割の計算時間は、プロセッサ数が 8 台からと 64 台からとで並列化の方法が変わるために階段状に減少している。一方、運動方程式の計算は、プロセッサ台数が増えるに従って並列化による加速が低下している。図 9 の communication で表される総通信時間は、プロセッサの台数とともに増大し

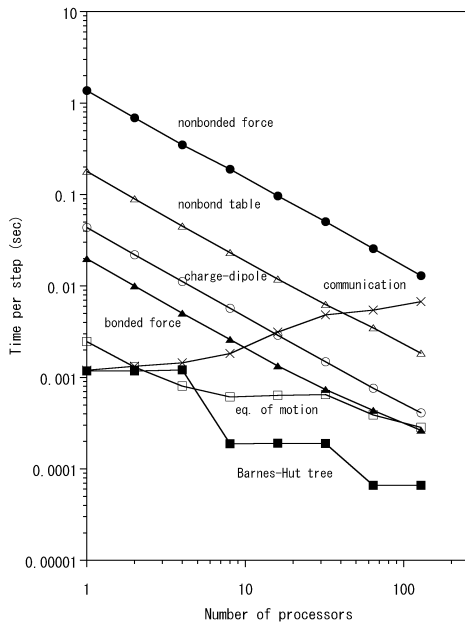


図9 図2の各サブプロセスごとの計算スピード

Fig.9 Calculation speed of each subprocess in Fig. 2.

ている。この部分は、さらなる高速化を行うときに解決すべき問題点となる。

図9の各処理について、ベクトル化による高速化をプロセッサ台数ごとに調べた結果を図10に示した。図10において、運動方程式の計算 (eq. of motion) と総通信処理 (communication) をのぞいたすべてのサブプロセスは、プロセッサ台数が増大してもベクトル化による加速が維持されていることが分かる。その理由は、これらのサブプロセスは2重ループからなっており、外側ループを並列化して内側ループをベクトル化しているために、ベクトル長が並列化によって短縮されないためである。

一方、運動方程式の計算と総通信処理は、プロセッサ台数の増加とともに、ベクトル化による加速が急激に落ち込む。その理由は、運動方程式の計算は、粒子数についての単純一重ループであり、それをプロセッサ台数で分割しているために、ループ長が急激に短くなるためである。ただし、運動方程式の計算時間は非常に短いために、ベクトル化による加速の低下は、全体の処理スピードに悪影響を及ぼさない。一方、総通信処理は、MPIの通信関数以外に前後のデータコピーを含むためにベクトル化による加速が見られる、しかし、データコピーに対して通信処理はプロセッサ台数とともに増大するために、ベクトル化による加速は低下する。以上のことから、図7に見られるような、全体のベクトル化加速がプロセッサ台数とともに低下す

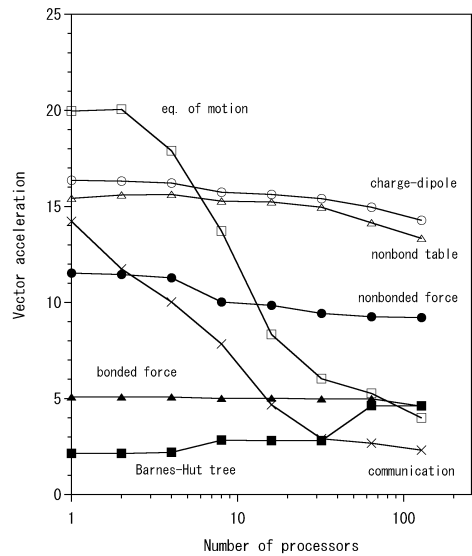


図10 図2の各サブプロセスのベクトル化による加速  
Fig.10 Vector acceleration of each subprocess in Fig. 2.

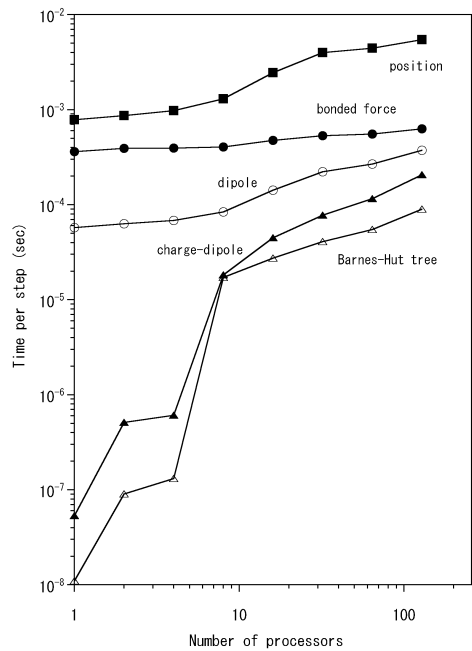


図11 図2の各通信処理のスピード

Fig. 11 Speed of each communication process in Fig. 2.

る原因は、ベクトル長の短縮が原因ではなくて、プロセッサ台数の増大とともに演算の時間に対して通信の時間が相対的に増大するためである。

図2に示したすべての通信処理ごとに、経過時間のプロセッサ台数依存性を調べた(図11)。通信処理のうちで時間のかかるのは、図2の(5)粒子の座標 (position)(2)結合性の力 (bonded force)(3.4)粒子の

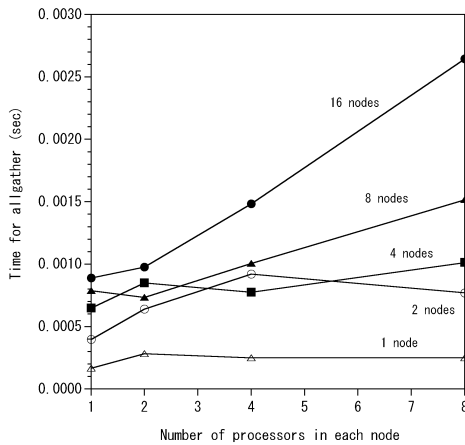


図 12 座標の通信の allgather のスピード

Fig. 12 Speed of allgather in the communication of positions.

双極子 (dipole) (3.6) セルの電荷と双極子 (charge-dipole) (3.2) 子セルと次セル (Barnes-Hut tree) の順番である。結合性の力の通信は、プロセッサ台数の増加とともに緩やかに増えるが、座標の通信は急に増大する。その結果、128 台のプロセッサにおいて、座標の通信処理には、その他の通信処理に比べて 10 倍以上の時間がかかっている。

座標の通信処理に含まれている allgather に要する時間を、図 12 に示した。ノード数とノード内の使用するプロセッサ台数を変えて (横軸)、通信時間をプロットした。128 台のプロセッサ (16 ノード、各ノード 8 プロセッサ) を用いた場合の、座標の通信処理の約半分は、allgather に要する時間であり、残り半分はデータコピーに要する時間である。allgather に要する時間は、1 ノード内 8 プロセッサでは 0.0003 秒、ノード内 1 プロセッサを 8 ノードの場合は 0.0008 秒、16 ノードでも 0.0009 秒であるにもかかわらず、ノード内 8 プロセッサを 16 ノードでは 0.0027 秒もの時間がかかる。128 台のプロセッサでの通信の際には、まずノード内の 8 プロセッサで通信を行ってから、次に 16 個のノード間で通信を行うことによって時間を短縮できるかもしれない。

## 6. 通常の PC との比較

地球シミュレータの性能評価に用いた計算 (図 6) とまったく同じ計算を、通常の PC クラスタで実行して計算スピードを評価した。COSMOS90 のコードは、MPI によって並列化されているため、MPI 通信ライブラリを実装した PC クラスタで、そのままコンパイルと実行ができる。このとき、PC のコンパイラは、ベク

トル化指示行を単なるコメント行として解釈する。筆者が自作した Beowulf タイプの PC クラスタを計測に用いた。次のような PC5 台を自作し、GbE スウィッチングハブで接続している。Chipset: intel7505, CPU: dualXeon 2.66 GHz (FSB: 533 MHz, L2: 512 KB, Memory: 3 GB DDR (PC2100)), OS: RedHat9.0, compiler: ifort 8.0, MPI Library: LAM7.0.4。

ヘモグロビンの計算スピードは、1, 2, 4, 8 プロセッサで、それぞれ 0.82, 0.69, 0.44, 0.17 sec/step であった。この計算スピードは、地球シミュレータのスカラー計算のスピードに匹敵する。すなわち、地球シミュレータの真価は、アプリケーションプログラムを高度にベクトル化して、高並列で利用しなければ発揮されない。一方、DNA 蛋白質複合体の計算スピードは、それぞれ 0.42, 0.26, 0.14, 0.09 sec/step であった。地球シミュレータのスカラー計算スピードよりもさらに数倍遅いが、実用化の域に達している。地球シミュレータは、中規模の蛋白質の計算時間を、さらに数十倍短縮できる。PC クラスタ上での MD シミュレーションの並列化加速性能は、リニアではなく複雑な様相を示している。興味深い結果であるが、その分析は本論文の主題と異なるので今後の研究課題とする。

## 7. 結 論

地球シミュレータにおいて、蛋白質の大規模シミュレーションをする際の、ベクトル化と並列化による加速性能を調べた。従来のスカラー型高並列計算機やベクトル型並列計算機で行うことができなかった大規模な蛋白質を高速にシミュレーションできる。ベクトル化と並列化による加速性能を調べた結果、以下のことが分かった。

- (1) 最も時間のかかる計算は、2 重ループであるので、並列化による分割をしてもループ長が短くならない。したがって、ベクトル化による加速は衰えない。単一ベクトルプロセッサのスーパーコンピュータでは 2 重ループを一重化して高速化したが、ベクトル型並列計算機ではその必要はない。
- (2) ベクトル化による加速と並列化による加速の両方を満足するアルゴリズムを採用することによって、理想的な高速化を達成できる。
- (3) 並列化による加速と比べて、ベクトル化による加速は、通信のオーバーヘッドをとみなさないために容易に達成できる。
- (4) 通信処理は、並列化による加速を低下させるだけでなく、ベクトル化されないためにベクトル化による加速をも低下させる。



(5) 単一ループを並列化してさらにベクトル化する際には、プロセッサ台数の増加とともにループ長が短縮され、ベクトル化による加速が急激に低下する。しかし、単一ループは、2重ループに比べて計算量が少ないため、このことによって全体の高速化は鈍らない。

### 参考文献

- 1) McCammon, J.A., Gelin, B.R. and Karplus, M.: Dynamics of folded proteins, *Nature*, Vol.267, pp.585-590 (1977).
- 2) 斎藤 稔, 佐谷野健二: 地球シミュレータにおける蛋白質のシミュレーションのベクトル化と並列化による高速化, HPCS2004, pp.81-86 (2004).
- 3) Sato, H., Tanaka, Y., Iwama, H., Kawakita, S., Saito, M., Morikami, K., Yao, T. and Tsutsumi, S.: Parallelization of AMBER molecular dynamics program for the AP1000 highly parallel computer, *Proc. SHPCC-92*, pp.113-120 (1990).
- 4) Saito, M.: Molecular dynamics simulations of proteins in water without the truncation of long-range Coulomb interactions, *Molecular Simulation*, Vol.8, pp.321-333 (1992).
- 5) 斎藤 稔, 佐谷野健二: 最新のベクトル型並列計算機による蛋白質のシミュレーションの高速化, JSPP2002, pp.179-180 (2002).
- 6) Barnes, J. and Hut, P.: A hierarchical  $O(N \log N)$  force-calculation algorithm, *Nature*, Vol.324, pp.446-449 (1986).
- 7) Makino, J.: Vectorization of a tree code, *J. Comput. Phys.*, Vol.87, pp.148-160 (1990).
- 8) Saito, M. and Sarai, A.: Free energy calculations for the relative binding affinity be-

tween DNA and  $\lambda$ -repressor, *Proteins*, Vol.52, pp.129-136 (2003).

- 9) Gao, J., Kuczera, K., Tidor, B. and Karplus, M.: Hidden thermodynamics of mutant proteins: a molecular dynamics analysis, *Science*, Vol.244, pp.1069-1072 (1989).
- 10) <http://www.scripps.edu/brooks/Benchmarks/>
- 11) Chen, E., Goldbeck, R.A. and Kliger, D.S.: Nano-second time-resolved spectroscopy of biomolecular process, *Annu. Rev. Biophys. Biomol. Struct.*, Vol.26, pp.327-355 (1997).

(平成 16 年 10 月 4 日受付)

(平成 16 年 12 月 8 日採録)



斎藤 稔 (正会員)

昭和 29 年生。昭和 60 年名古屋大学大学院理学研究科物理学専攻博士課程修了。昭和 63 年蛋白質工学研究所主任研究員。蛋白質の分子動力学シミュレーションプログラム (COSMOS90) を開発。これによって、水中の蛋白質を長距離クーロン力をカットオフせずにシミュレートすることを世界に先駆けて可能にした。平成 8 年新情報処理開発機構主任研究員。アプリケーションプログラム (COSMOS90, AMBER, Barnes-Hut treecode) を並列計算機上で高速化。平成 10 年より弘前大学理工学部教授。情報科学, 生命科学, 物理化学の各分野にまたがった研究に従事。理学博士。日本生物物理学会, 日本蛋白質科学会, 日本物理学会, 日本化学会各会員。