ハイパフォーマンス
コンピューティング 67－6
（1997. 8. 19）

蛋白質立体構造データベース (PDB) の代表蛋白質決定システムの並列化

野口 保[†] 秋山 泰[†] 鬼塚 健太郎[†]
斎藤 稔[†] 安藤 誠[†] 志澤 由久[††]

蛋白質立体構造データベース (PDB) は、近年のX線結晶解析やNMRによる構造解析技術の進歩により、その内容は現在５８００エントリー (2.4Gbytes) を越え、今後もさらに増え続けると予想されている。しかしながら、冗長性やデータの不完全性のためにＰＤＢ全てのエントリーが蛋白質の立体構造の解析に適しているとは言えず、何らかの基準で代表蛋白質を決定する必要がある。この代表を決定するには、各エントリーの内容を調べ、解析に適さない質の悪いデータを除去したり、各エントリーに対して他のエントリーが配列的および立体構造的に類似の蛋白質かどうかを調べあげるなど、膨大な計算が必要となる。さらに、ＰＤＢの増加を考慮すると、今後も急速に計算量が増えていく課題である。

我々は、これまで開発してきたＰＤＢの代表蛋白質決定システムを、MPIライブラリを用いて並列化し、３種類の並列計算機上でその速度性能を調べた。本研究で実装した並列版では、従来の約10倍の高速化を実現し、およそ２週間を必要としていた代表蛋白質決定処理を２日間で実行できるようになった。

Parallelization of the automatic determination system for representative protein chains of the Protein Data Bank (PDB)

Tamotsu Noguchi,[†] Yutaka Akiyama,[†] Kentaro Onizuka,[†]
Minoru Saito,[†] Makoto Ando[†] and Yoshihisa Shizawa [††]

The Protein Data Bank (PDB) is a rich library of atomic-coordinate data of biological macromolecules. The PDB entries has been increasing rapidly by the improvement of X-ray crystallography and NMR experimental techniques, and the number of current entries is more than 5,800 (2.4Gbytes), though not all entries are competent for the purpose of computational protein structure analysis. A lot of entries have insufficiently-refined coordinate data.

Thus we have developed a representative chain database PDB-REPRDB, and in this paper we report the MPI- parallelization of our automatic construction system for PDB-REPRDB. Performance evaluation on three parallel computers is also reported. Now that a calculation of a representative set can be done within 2 days rather than 2 weeks, with 10-folds speed-up achieved in this study.

1. はじめに

ＰＤＢ (Protein Data Bank)[1] は、米国のブルックヘブン国立研究所から提供されている蛋白質立体構造データベースで、X線やNMRなどの構造解析により明らかにされた生体高分子（蛋白質, DNA, RNA など）の立体構造が、その解析結果ごとに１ファイル（エントリー）の形式で登録されている。

近年のＸ線結晶解析やNMRによる構造解析技術の進歩により、そのデータ量は１９９１年ごろから急激に増加し、１９９７年４月版で５８００エントリー (2.4Gbytes) を越え、さらに増え続けている（図１）。

しかしながら、そのエントリーの多くは配列と立体構造がともに類似している近縁の蛋白質である。近縁蛋白質の基準として、たとえば、

- 配列の相同性基準：ID%（配列を重ね合わせた際の同一アミノ酸残基の比率）$\geq$ 75%、
- 立体構造の相同性基準：（構造を重ね合わせた際の原子間距離の最大値）$\leq$ 10.0 Å、

を採用すると実に全エントリーの85%は他の蛋白質と近縁関係にある。またＰＤＢデータは、実験方法の差異、分解能やリファインメント☆の度合いなどによってデータの質（信頼度）が様々である。ＰＤＢデータを利用する場合、類似のデータがあれば、より質の良いデータを利用した方が、解析誤差を低く抑えられる。

† 新情報処理開発機構
Real World Computing Partnership
†† （株）情報数理研究所
Information and Mathematical Science Laboratory, Inc.

☆ リファインメント (refinement): 実験データをもとに立体構造を構築していく段階で、実験データと矛盾なく、かつエネルギー的により安定な構造を力学計算により決める処理。

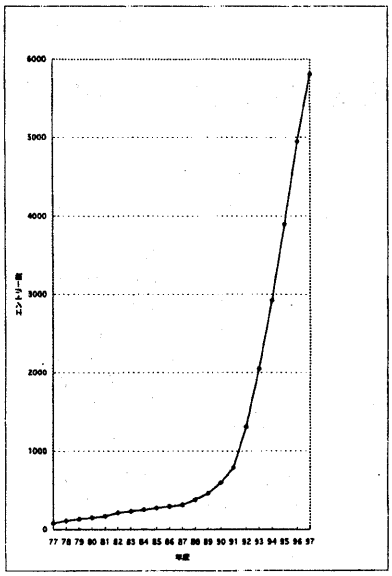

図1　蛋白質立体構造データベース (PDB) のエントリー数の推移

立体構造が既に明らかな蛋白質との配列上の類似性をもとに、未知の立体構造を予測する統計的立体構造予測法の研究では、前述の近縁蛋白質を無視して統計をとると情報の偏りを生じてしまい、誤った予測をする可能性が高い。そのため、一定の基準（たとえば、配列の相同性：ID% $\geq$ 30%）で近縁蛋白質の代表を選んでおくことが、この種の研究を進める上できわめて重要である。このような用途での“代表点”は、比較的遠い関係の蛋白質もカバーする“半径の大きな”ものとなる。

他方、よく類似した配列をもとに蛋白質の未知立体構造をモデリングしたい場合には、別の基準（たとえば、配列の相同性：ID% $\geq$ 95%）での近縁蛋白質の代表点を決めておくことが有益である。この場合の“代表点”は“半径の小さな”ものとなり、よく類似した蛋白質の中で良質の構造が選ばれる。これにより、近接した良質な立体構造を選んでモデリングを始めることができる。

このような需要のもとにHobohmら[2),3)]は、配列間の相同性を考慮して、ＰＤＢの代表蛋白質チェインを決定する方法を提案し、“PDB_SELECT”[4)]として公開している。現在この代表は配列の相同性：ID% $\geq$ 25% ～ 95%まで10%刻みの基準で用意されており、蛋白質立体構造の研究者の間で広く用いられている。

しかし一方で、たとえ配列の相同性が高い蛋白質であっても、立体構造を重ね合わせた時に、部分構造が大きく異なることがある。このような局所的構造のバラエティを残して、研究用のデータセットを作成したい場合には、従来からの配列の相同性だけを基準とする方法では不十分である。そこで我々は、配列の相同性が高いチェイン同士を比較し、部分的に立体構造が異なるチェインは別の代表点とする“PDB-REPRDB” V.1.0[5)]を作成した。ただし、この時点での選定作業には、多くの手作業が残されていた。

その後、ＰＤＢのエントリー数の急激な伸びに対応するため、ＰＤＢの代表蛋白質決定システムの自動化（逐次版）を進めてきた[6)]。現在は、PDB-REPRDB V.2.1（配列の相同性：ID% $\geq$ 75%, 80%, 90%）[6)]を公開している。しかし、様々な研究用途に使うためには、配列と構造の相同性のしきい値をもっと広範囲に変えて代表蛋白質を決定していく必要がある。

しかし最近のＰＤＢに対し、逐次版システムは代表決定に約2週間の計算時間を要しており、もしも配列の相同性：ID% $<$ 25% ～ 95%まで10%刻みの8通り、構造の相同性：5 Å, 10 Å, ∞ Å(構造差を認めず)の3通りの基準を組み合わせて、$8 \times 3 = 24$通りの代表蛋白質セットを計算したいような場合には、その計算時間は膨大なものとなる。今後のＰＤＢの増加も考慮しつつ、年4回行なわれているＰＤＢの新版リリースに迅速に対応するには、処理の大幅な高速化が必須である。

そこで本研究では、このＰＤＢ代表蛋白質決定システムの並列化を行ない、処理速度の高速化を図った。

2. ＰＤＢの代表蛋白質決定システム

PDB-REPRDBは、ＰＤＢをもとに、以下の手順で作成される（図2）。

2.1 不適切なデータの除外

ＰＤＢのエントリーをまずチェイン☆単位に分離したのち、下記に該当するデータを取り除く。

a)　DNA と RNA データ
b)　NMR で解析されたデータ
c)　理論計算で求められたモデルデータ
d)　チェインの長さが短いデータ ($l < 40$ 残基)
e)　全ての残基において主鎖座標がないデータ
f)　全ての残基において側鎖座標がないデータ
g)　リファインメント処理が行われていないデータ

2.2 データの質による順位付け

ＰＤＢデータのチェインごとに下記のプライオリティーで順位付けを行う。初めに準備として、分解能が3.0 Å 以下かつＲファクターが0.3以下のチェインをクラスＡと分類し、それ以外のチェインをクラスＢとする。クラスＡのデータはクラスＢのデータよりも上位とする。それぞれのクラスのチェインは、まず分解能、次にＲファクターの小さい順に並び替えられる。分解能、Ｒファクターがともに等しい場合は、さらに下記の項目を順に調べて順位付けを行う。

(1)　チェイン・ブレイク☆☆の数 (少ないほど上位)

☆ 蛋白質が単数の／複数のポリペプチド鎖で構成されるときの各鎖
☆☆ チェイン・ブレイク (chain break): ＰＤＢの座標において、チェインの途中で座標を決定できなかった原子が存在したためチェインが切れたように見える状態。または、リファインメントが不十分なため、主鎖の原子間距離が異常に離れた状態。

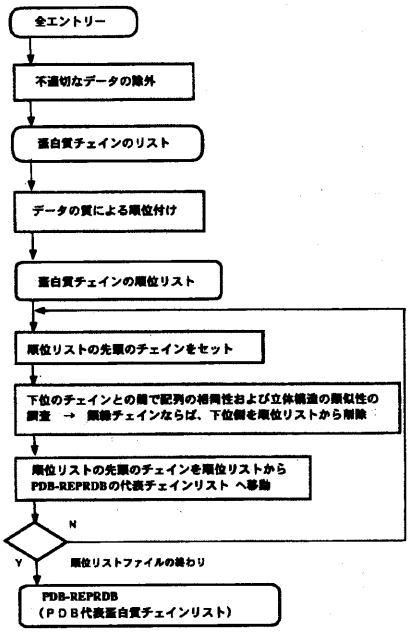

図2　PDBの代表蛋白質決定システムの流れ

(2)　標準的なアミノ酸残基種以外の残基の数 (少ないほど上位)
(3)　主鎖原子の座標を欠く残基の数 (少ないほど上位)
(4)　側鎖原子の座標を欠く残基の数 (少ないほど上位)
(5)　野生型か変異体か (野生型が上位)
(6)　複合体かどうか (複合体でないものが上位)
(7)　チェイン名のアルファベット順
(例 1MCD < 1MCE, 5AT1A < 5AT1C)

2.3　類似蛋白質チェインの検索および代表蛋白質チェインの決定

上記の処理によりデータの良質度でソートされたリスト（順位リスト）が得られるので、リスト上位のものを優先しながら、互いに近縁関係がないような代表チェインを選び出し、選択されなかったチェインについては、どの代表に近いかでグループ分けを行なう。具体的には、まず上位のチェインのアミノ酸配列をキーにして、それ以下のチェインの配列相同性をDP(動的計画法)を用いたペアワイズアライメントの手法[7]で調べる。その相同性がしきい値以上であれば、下位側は消去しても良いと考えられる。しかし、我々の手法では、ここでさらに両者の C_α 原子をもとにチェイン全体の骨格の重ね合わせを行い、全ての原子間のズレがしきい値以下であり、立体構造の差異がないと認められる時に初めて下位側をリストから削除し、近縁蛋白質チェインとして、代表点（上位側）と同じグループのリストに加える。

この処理を順にリストの最後まで行うことにより、近縁グループおよびその代表蛋白質チェインを決定する。立体構造の比較にはKabschによる最小2乗フィット法[8]を採用し、重ね合わせた原子間距離の最大値を求める。この基準により、構造全体間の距離（r.m.s.d値）では検出できなかった部分構造の違いを検出できる。

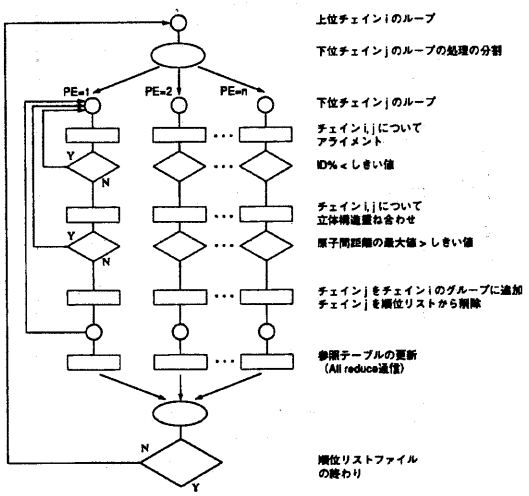

図3　並列PDB代表蛋白質決定システムにおける計算の流れ

3.　代表蛋白質決定システムの並列化実装

MPIライブラリを利用して、PDBの代表蛋白質決定システムの並列化を行なった。

逐次版システムにおいて、処理時間の90%以上を要していた“類似蛋白質チェインの検索および代表蛋白質チェインの決定”の部分（図2におけるループ）の内部において、上位側チェイン i が与えられたとき下位側チェイン j との比較処理は各 j について同時に行なえることから、これをSPMD方式で並列化した（図3）。

順位リストの各チェインが近縁蛋白質として削除された状態か、未削除かを記録する参照テーブルを用意する。この参照テーブルをもとに比較を行なうべきチェインが決められ、以下の処理が並列実行される。

並列に処理されるのは、配列間アライメント、立体構造重ね合わせ、および参照テーブルの更新である。蛋白質チェインのリストと全配列データは、n 台のプロセッサの全てに配布しておく。

上位側チェイン i と比較すべき下位側チェイン j の各PUへの分担法は計算の当初から静的に決められており、チェイン番号にしたがいブロックサイクリック的に対応づけられる。すなわち m 本のチェイン c_0 から c_{m-1} があるとき、第 i 番目のチェイン c_i を担当すべきプロセサの番号 p $(1 \le p \le n)$ は、

$$p = \left(\left\lfloor \frac{i}{k} \right\rfloor \bmod n\right) + 1 \qquad (1)$$

で決定される。ただし k はブロック幅（今回は1）、n は使用するプロセサ台数とする。

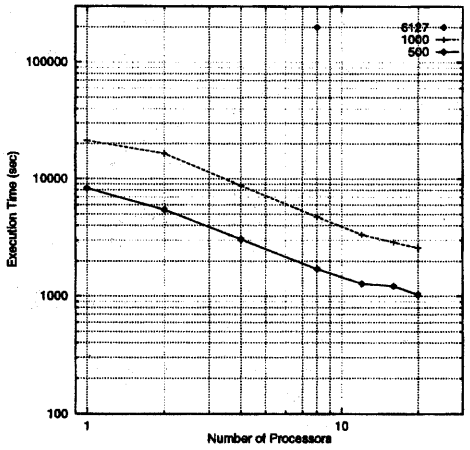

図4 SPARCcenter 2000E 上での処理時間

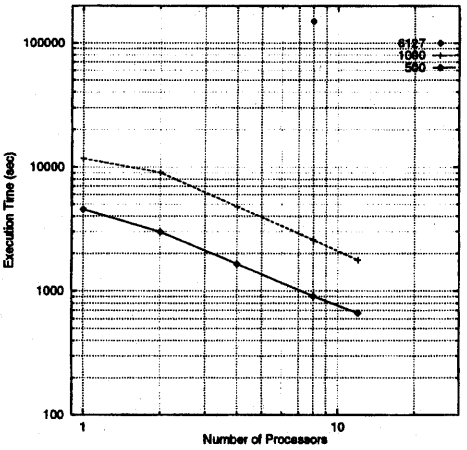

図5 Power Challenge 上での処理時間

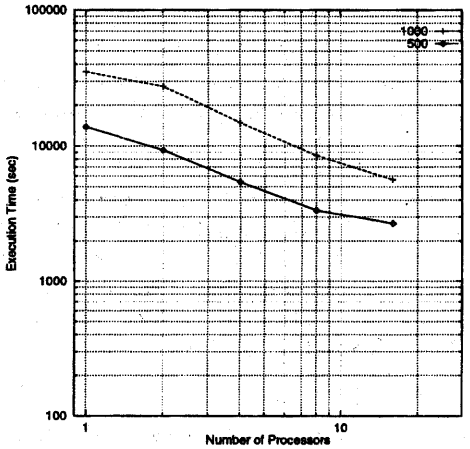

図6 SR2201 上での処理時間

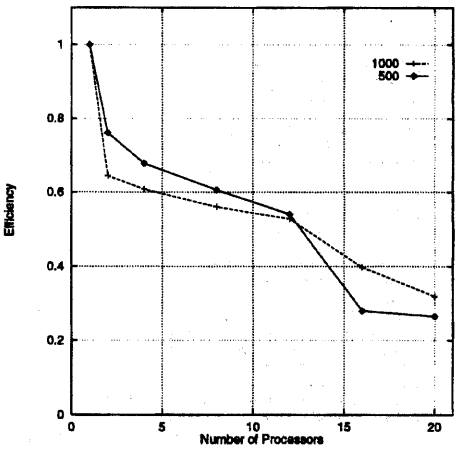

図7 SPARCcenter 2000E 上での対逐次効率

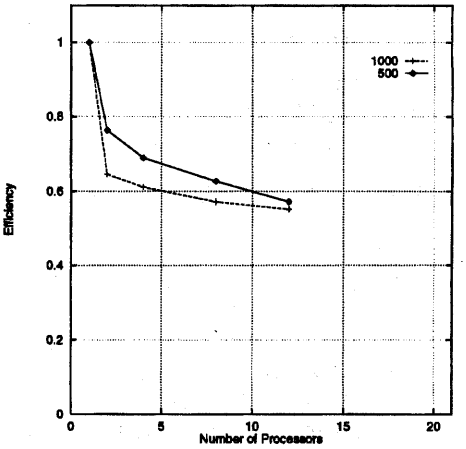

図8 Power Challenge 上での対逐次効率

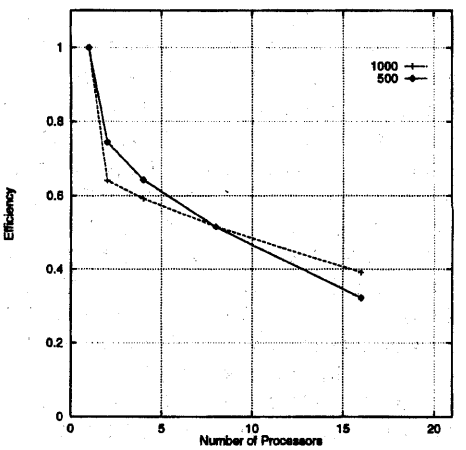

図9 SR2201 上での対逐次効率

配列間アライメントで用いる各チェインの配列データは各PUのメモリ上に保持し、立体構造重ね合わせで用いる原子座標データは、必要に応じてPDBファイルから読み込むことにした。立体構造重ね合わせが行なわれるのは、アライメントの結果、相同性が高かった時のみであり、その実行の割合はアライメント約500回に対し1回程度である。また原子座標のデータ量は、C_α原子部分だけでも大きい（約120Mbytes）ため、各PUのメモリには配列データ（約10Mbytes）のみを置いた。

必要な通信は、最初に参照テーブルと全配列データを各PUにブロードキャストし、以降は図3の上位側チェインiのループが終了するごとに、各PUで削除したチェイン名を収集して、参照テーブルの内容を更新して再びブロードキャストすることである。

計算量については、配列間アライメントは配列長の二乗のオーダ、重ね合わせは一乗のオーダであるが、蛋白質の配列長のバラツキが大きいため、計算時間は配列ペアごとに大きく変わる。システム全体では、チェイン数mに対して二乗のオーダとなる。配列および構造の相同性のしきい値、およびデータベースの内容によって、削除されるチェイン数が異なり、計算量が変動する。

4. 性能評価

表1に掲げる3種類の並列計算機上で、並列版ＰＤＢ代表蛋白質決定システムの速度性能を評価した。使用したＰＤＢは、リリース#７８ (エントリー数: 4873, 全チェイン数: 8870, 順位リストに残るチェイン数：6127) である。チェイン数による性能の違いを評価するため、順位リストの上位500, 1000, 6127本のチェインからなる3つのサブセットを作り、性能評価に利用した。

図4にSPARCcenter 2000Eでの処理時間、図7に対逐次効率を示す。順位リストのチェイン数が500本の場合と1000本の場合とで、ほぼ同様の性質を示している。1000本の場合の方がやや並列効率が悪くなっている理由は、計算の後半に負荷分散が悪化して（6.1節参照）、チェインiのループごとの同期で待たされているPUが増えるためと考えられる。

図5にPower Challengeでの処理時間、図8に対逐次効率を示す。並列効率については、SPARCcenterと同様の傾向が得られた。各PUあたりSPARCcenterの約2倍の演算性能を示しており、絶対性能としては、Power Challengeを12PU利用した場合が、今回の実験中では最も高かった。12PUを利用した場合、対逐次効率は55%前後で、約7倍の台数効果を得ている。（6.2節で述べる、配列データのオンメモリ化の効果を含めると、従来法の約10倍の速度である。）

図6にSR2201での処理時間、図9に対逐次効率を示す。PUあたりの得られた性能が他の計算機よりも悪く、PowerChallengeの0.3倍程度の性能しか出ていない。原因を調査したところ、C++言語で記述された各種ルーチン（6.3節参照）の処理時間が、他の計算機に比べて極端に遅いことがわかった。C++コンパイラの性能に由来するものではないかと疑われる。SR2201は利用可能なPU数が多いため、しきい値の設定が異なる複数の計算をタスク並列で実行させる (たとえば4PU×24試行) といった利用も、我々は期待している。

表1　並列ＰＤＢ代表蛋白質決定システムの性能評価を行なった機種

機種	PUチップ	PU数	主記憶	通信
Sun SPARC center2000E	SuperSPARC II 85MHz	20	5GB 共有	MPI
SGI Power-Challenge	R10000 194MHz	12	2GB 共有	MPI
Hitachi SR2201	PA-RISC1.1+PVP-SW 150MHz	256	64GB 分散	MPI

5. ＰＤＢ代表蛋白質の公開

本システムの処理結果であるＰＤＢの代表蛋白質は、PDB-REPRDB[6] として、図10の例のようにWWWで公開している。代表チェインを表形式で示しており、選ばれた代表蛋白質のエントリー名とチェインIDおよび残基数、分解能、Ｒファクター、実験方法、主鎖原子の座標がそろっている残基数、側鎖原子の座標がそろっている残基数、EC（酵素）番号、蛋白質名が記されている。また、エントリー名とチェインIDの部分は、ＰＤＢとホットリンクしており、クリックするとそのＰＤＢエントリーの内容が表示される。またRasMolプログラムを用いた立体構造のグラフィック表示もできる。

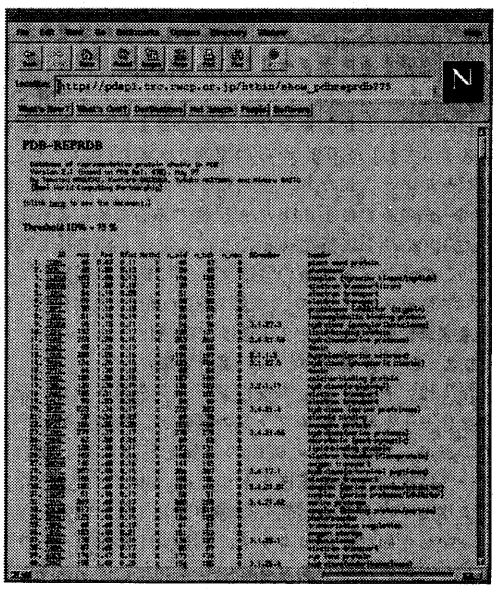

図10　WWW上のPDB-REPRDB

6. 考　　察

6.1 負荷分散

配列間の相同性検索の部分において、下位側チェインを各PUに分担させる方法には改良の余地がある。現在はチェイン番号から静的に担当すべきPUが割り振られている。この方法では、処理の後半で削除フラグの立った配列が多くなると、負荷に偏りが生じて、処理が早く終るPUと遅く終るPUの差が出る。今回は全配列データが各PUにあらかじめロードされているので、削除されずに残っている配列をそのつどフラグで確認して、各PUが均等に分担する方法に改良する予定である。

各PUへの割り当てを考える際に、担当本数だけでなく、アラインメントの処理時間を大きく左右する配列長に注目して分割していく方法[9]が、データベース検索の分野で提案されている。しかし今回のケースでは、相同性の結果に応じて立体構造の比較処理が発生し、PDB座標ファイルをアクセスすることから、各ステップの開始時に良い負荷分散を静的に計画することは簡単ではない。今後は、より動的な負荷分散法も検討したい。

なお、リストが短くなるにつれて粒度が小さくなり、通信のオーバーヘッドが生じているが、外側のループは単純には並列化できない（結果によりロールバックを生じる）ため、より詳細な検討が必要である。

6.2 配列データのオンメモリ化

本システムは大量のPDBファイル間で相互の類似度を調べる処理であることから、最もナイーブな実装では、順位リストチェイン数の二乗の回数(3600万回)のファイルアクセスが発生することになる。配列比較を行なうのに、複雑なPDBファイルを毎回読み込むのは非効率である。そこで従来の逐次版でも、類似検索に必要な配列部分をあらかじめPDBから読みだし、一つに統合した配列ファイルを作成し、そこから配列を読むようにしていた。しかし単一ファイルのままでは、並列版のシステムでは、各PUからの配列ファイルへのアクセスがボトルネックになり、並列効率が上がらなかった。配列データのみでは10Mbytes程度であるため、これを各PUのメモリ上に持たせることにした。4節で示したのは、このオンメモリ版の性能データである。従来のファイルアクセス版に比べて、SPARCcenterの場合で、実行速度が全体で1.8～2.0倍ほど高速化されている。

6.3 C++プログラムの実行効率

我々は、PDBなどの蛋白質データを扱うための統合的なC++クラスライブラリの開発と、その並列化を進めており、今回もその機能の一部を利用した。蛋白質データは複雑な階層構造をもつため、オブジェクト指向により、プログラム開発の効率が著しく向上する。

しかし4節で述べたように、SR2201ではC++プログラムの実行性能が高くはないことから、マシン本来の性能をうまく引き出せていない点が課題となっている。

7. ま と め

蛋白質立体構造データベース（PDB）の代表蛋白質決定システム[6]を、MPIライブラリを用いて並列化した。並列化(～7倍、12PUの場合)と配列のオンメモリ化(1.8～2.0倍)の効果で、以前の逐次版システムに比べて約10倍の速度が達成され、これまで約2週間かかっていた計算を2日間で行なうことが可能になった。

従来は、代表チェインの選出は手仕事で行なわれており、様々に条件を変えて代表リストを作成したり、研究者の注文に合わせて即時に計算することなどは全く不可能であった。本研究での並列化をはじめとする高速化、自動化により、現在それが可能になりつつある。

PDBデータベースは急激に増大しているため、今後さらに並列化を進めて、速度向上を目指す必要がある。

参 考 文 献

1) Bernstein, F. C.; Koetzle, T. F.; Williams, G. J. B.; Meyer, E. F., Jr.; Brice, M. D.; Rodgers, J. R.; Kennard, O.; Shimanouchi, T.; and Tasumi, M. The Protein Data Bank: A Computer-based Archival File for Macromolecular Structures. *Journal of Molecular Biology* **112**, pp.535-542 (1977). `http://www.pdb.bnl.gov/`

2) Hobohm, U.; Scharf, M.; Schneider, R.; and Sander, C. Selection of a representative set of structures from the Brookhaven Protein Data Bank. *Protein Science* **1**, pp.409-417 (1992).

3) Hobohm, U.; and Sander, C. Enlarged representative set of protein structures. *Protein Science* **3**, pp.522 (1994).

4) Hobohm, U.; and Sander, C. PDB_SELECT: Representative list of PDB chain identifiers. `http://www.sander.embl-heidelberg.de/pdbsel/explanation.html`

5) Noguchi, T. PDB-REPRDB ver.1.0 served on GenomeNet WWW Server. (1995). `http://www.genome.ad.jp/htbin/show_pdbreprdb`

6) Noguchi, T.; Onizuka, K.; Akiyama, Y.; and Saito, M. PDB-REPRDB:A Database of Representative Protein Chains in PDB (Protein Data Bank). *Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology.*, pp.214-217 (1997) `http://www.rwcp.or.jp/lab/mpap/`

7) Needleman, S.B. and Wunsch, C.D., A general method applicable to the search for similarities in the amino acid sequences of two proteins. *Journal of Molecular Biology*, **48**, pp.443-453 (1970).

8) Kabsch, W., A discussion of the solution for the best rotation to relate two sets of vectors. *Acta Cryst.* **A34**, pp.827-828 (1978).

9) Yap, T. K.; Frieder O.; and Martino, R. L., Parallel Homologous Sequence Searching in Large Databases. *Proceedings of the Fifth Symposium on the Frontiers of Massively Parallel Computation*, pp.231-237 (1995).