Ninf-1/Ninf-G を用いたNMR蛋白質立体構造決定のための遺伝アルゴリズムのグリッド化

小野 功[†1] 水口尚亮[†1] 中島直敏[†1]
小野典彦[†1] 中田秀基[†2,†3] 松岡 聡[†3,†4]
関口智嗣[†2] 楯 真一[†5]

核磁気共鳴法 (NMR) は，ポストシーケンスにおける最重要課題の一つである蛋白質立体構造解析の有望な手段である．しかし，専門家でさえ，一つの蛋白質のデータ解析に数ヶ月程度の試行錯誤が必要なことが深刻な問題となっている．これに対し，Ono らは遺伝アルゴリズム (GA) に基づくデータ解析の自動化手法を提案し，13 残基程度の小規模な問題において比較的良好な性能を得られたと報告している．しかし，実際に扱わなければならない問題規模は数十残基から二百残基程度である．たとえば，78 残基の場合，Pentium III 1.4GHz のシングル CPU の PC では，計算が終了するまでに約 200 日程度の時間がかかってしまうことが見積もられており，高速化が望まれている．本論文では，ミドルウェア Ninf-1 および Ninf-G を用いて Ono らの提案した GA をグリッド上で並列化したシステム（NMR 蛋白質立体構造決定のためのグリッド向け GA システム）を提案する．5 サイト/1,196CPU から構成されるグリッドテストベッド上で，提案システムの性能評価を行う．

Gridifying A Genetic Algorithm for NMR Three-Dimensional Protein Structure Determination by Using Ninf-1 and Ninf-G

Isao Ono,[†1] Naoaki Mizuguchi,[†1] Naotoshi Nakashima,[†1]
Norihiko Ono,[†1] Hidemoto Nakada,[†2,†3] Satoshi Matsuoka,[†3,†4]
Satoshi Sekiguchi[†2] and Shin-ichi Tate[†5]

Nuclear Magnetic Resonance (NMR) spectroscopy is a promising method for the three-dimensional structure determination of proteins that is one of the most important problems in post-sequence era. This method has a serious problem that it takes several months for experts to analyze the data of only one protein. In order to remedy the problem, Ono *et al.* have proposed an automatic method based on a genetic algorithm (GA) for analyzing the data and determining the three-dimensional structures of proteins and reported that they had good results on a small-size problem whose size was 13 residues. However, actually, the size of proteins that we have to handle is from several dozens of residues to about 200 residues. For example, in case of 78 residues, it is expected that it takes about 200 days to perform the GA if a single-CPU PC with Pentium III 1.4GHz is used. In this paper, we parallelize the GA on a grid by using Ninf-1 and Ninf-G that are middelwares. We call the GA parallelized on a grid *the Grid-Oriented GA system for NMR three-dimensional protein structure determination*. We examine the performance of the proposed system on a grid testbed consisting of five sites/1,196 CPUs.

†1 徳島大学
The University of Tokushima
†2 産業技術総合研究所
National Institute of Advanced Industrial Science and Technology
†3 東京工業大学
Tokyo Institute of Technology
†4 国立情報学研究所
National Institute of Informatics
†5 生物分子研究所
Biomolecular Engineering Research Institute

1. はじめに

ポストシーケンスにおいて蛋白質立体構造解析は最重要課題の一つであり，核磁気共鳴法 (NMR) は有望な構造解析技術の一つである．NMR による構造解析の過程は，1) 連鎖帰属，2) NOE 帰属および立体構造決定，の２つのフェーズに大別されるが，後者が律速となる．現状の NOE 帰属および立体構造決定のフェーズでは，高度な専門知識と豊富な経験に基づき，試行

錯誤的に NOE シグナルを ^{1}H ペアに帰属し，徐々に立体構造を構築していく作業が行われているため，専門家でも１つの蛋白質の立体構造決定に数ヶ月を要する．そのため，自動化および高速化技術の開発が強く望まれている[6]．

Ono らは，観測された NOE シグナルを満たす立体構造を遺伝アルゴリズム (Genetic Algorithm; GA) により探索することにより，NOE 帰属および立体構造決定の自動化を行う手法を提案している[5]．本手法をアミノ酸残基数 13 のベンチマーク問題に適用したところ，専門家とほぼ同じ立体構造を求めることに成功した．しかし，実際に扱う必要がある蛋白質の規模は，数十残基から二百残基程度である．たとえば，78 残基の場合，Pentium III 1.4GHz のシングル CPU の PC では，計算が終了するまでに約 200 日程度かかると見積もられており，高速化が緊急の課題である．

GA は，複数の解候補をサンプリングすることを膨大な回数繰り返しながら探索を進めていくため，本質的に並列化が可能である．Ono らの手法[5] において，計算時間の大半を占めているのは立体構造の評価計算であることから，並列化により大幅な高速化が期待できる．上述の 78 残基の場合，数百〜1,000CPU 程度の並列計算機上で並列実行できれば，数時間以下の現実的な時間で計算を終了できると考えられる．

グリッドは，地理的に分散した複数拠点の PC クラスタなどの計算資源を，インターネット越しに相互接続して並列処理を行うことにより，一拠点の計算資源では実行困難な大規模計算を可能にする次世代並列計算プラットフォームとして注目されている．近年，PC は安価に供給されており，10〜100 ノードの PC クラスタの構築は比較的容易になってきている．そのため，数〜十数の研究拠点が計算資源をシェアすることにより，かなり大規模なグリッド計算環境を構築することが可能であると考えられる．

本論文では，ミドルウェア Ninf-1[4),10)] および Ninf-G[4),9)] を用いて Ono らの GA[5)] をグリッド上で並列化したシステム（NMR 蛋白質立体構造決定のためのグリッド向け GA システム）を提案する．また，産総研グリッドセンター，東工大松岡研究室，東工大合田研究室，東京電機大藤沢研究室，徳島大小野研究室の 5 サイトに分散配置された 1,196CPU から構成されるグリッドテストベッドを用いて，提案システムのノード数に対するスケーラビリティ，耐障害性の評価を行う．さらに，実際に 78 残基の問題において提案システムを最適化計算が終了するまで実行し，数時間以下の現実的な時間で正常に計算を終了できることを示す．

2. NMR 蛋白質立体構造決定のための遺伝アルゴリズム[5)]

2.1 NMR 蛋白質立体構造決定問題

蛋白質は複数のアミノ酸がペプチド結合により鎖状に結合したものである．通常，蛋白質は，結合周りで回転することにより折りたたまれエネルギー的に安定な立体構造をとる．結合周りの回転角を二面角とよぶ．

蛋白質を溶液に溶かし，NMR 装置にかけると，ある特別な測定条件において，NOE と呼ばれるシグナルを数多く測定することができる．NOE は，お互いの距離が十分に近い (5 Å程度）水素原子核 ^{1}H のペアから観測される．NOE シグナルを対応する ^{1}H ペアに帰属することにより，^{1}H ペア間の距離制約を得ることができる．したがって，NOE 帰属により得られた膨大な数の距離制約を用いることにより，蛋白質の立体構造を決定できる．しかし，実際には，測定装置の分解能や誤差の問題から NOE を対応する ^{1}H ペアに一意に割り付けることは困難である．

Ono らは，上述の NOE 帰属の過程を，観測 NOE と予測 NOE の一致度を最大化し，かつ，原子の重なり具合を最小化するような立体構造を発見する最適化問題として定式化することを提案している[5)]．ここで，予測 NOE は評価対象の立体構造において原子間距離が 8 Å以下の ^{1}H ペアを列挙したものであり，観測 NOE を全てカバーしているとき，観測 NOE と予測 NOE の一致度は最大値（＝観測 NOE の数）をとる．原子の重なり具合は，評価対象の立体構造において原子同士の衝突がないとき最小値（＝ 0）をとる．

2.2 遺伝アルゴリズムに基づく解法

遺伝アルゴリズム (Genetic Algorithm; GA)[2)] は，自然界の生物の進化過程を模倣した最適化の枠組みである．GA の設計項目は，コード化/交叉・突然変異設計と世代交代モデル設計に大別される[7)]．以下，Ono らの手法[5)] の説明を簡単に行う．

コード化はアミノ酸残基の二面角を要素とする実数ベクトルを採用し，交叉は一様交叉 (Uniform Crossover; UX) を採用している．UX は，2 つの親個体が与えられたとき，50 %の確率で実数ベクトルの各変数を入れ換える．突然変異は，各変数について 1%の確率で，-1.0^o〜 $+1.0^o$ の一様乱数を加える．

世代交代モデルは，Minimal Generation Gap (MGG)[8)] に基づくモデルを採用している．図 1 にその概要を示す．アルゴリズムの詳細を以下に示す：

1．初期集団の生成: ランダムに複数個の個体を生成，評価値を計算し，それらを初期集団とする．

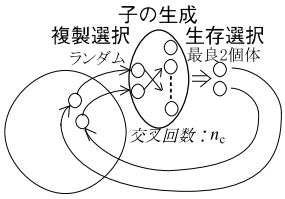

図 1 Minimal Generation Gap (MGG) モデル
Fig. 1 Minimal Generation Gap (MGG) model

2. **複製選択:** 集団からランダムに交叉のための 2 つの親を選択する．
3. **子の生成:** ステップ 2 で選択された両親に対し，交叉 UX を n_c 回適用し，子を $2n_c$ 個生成する．生成された子個体に対し，突然変異を適用する．
4. **子の評価:** ステップ 3 で生成された全ての子個体の評価値を計算する．
5. **生存選択:** 両親と生成された全ての子を合わせた個体集合から最良 2 個体を選択する．
6. **世代交代:** 選択された最良 2 個体を集団中の両親と置き換える．
7. 停止条件が満たされるまで，ステップ 2 から 6 を繰り返す．

3. NMR 蛋白質立体構造決定のためのグリッド向け GA システム

3.1 想定するグリッド計算環境

本論文で想定するグリッド計算環境を図 2 に示す．本グリッド計算環境は，ユーザー端末と各拠点に配置された複数の PC クラスタから構成される．ユーザー端末は，グローバル IP を持ち，外部から直接アクセス可能である．PC クラスタは，ゲートウェイと複数の計算ノードから構成される．ゲートウェイは，グローバル IP を持ち，外部から直接アクセス可能である．計算ノードは，グローバル IP またはプライベート IP をもち，同じ PC クラスタのゲートウェイおよび計算ノードからアクセス可能である．

3.2 NMR 蛋白質立体構造決定のためのグリッド向け GA システムの要件

前節のグリッド計算環境で動作する NMR 蛋白質立体構造決定のためのグリッド向け GA は，以下の要件を満たすべきであると考えられる：

セキュリティ: 強力なユーザー認証機構，通信路の暗号化機構が必要である．異なる組織が管理している計算資源へインターネットを介してアクセスしなければならないためである．

スケーラビリティ: 1,000CPU 程度までのスケーラビリティが必要である．実際の規模の問題を現実的な時間で解くためには，1,000 倍程度の高速化が必要なためである．

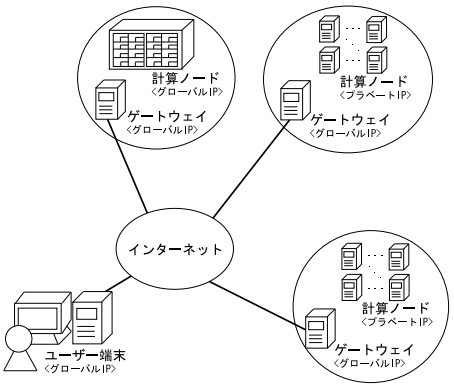

図 2 本論文で想定するグリッド計算環境
Fig. 2 Grid computaion environment assumed in this paper

耐障害性: 一部の計算ノードに障害が起こった場合でも，全体の計算は継続されることが必要である☆．また，ユーザー端末に障害が起こった場合でも，途中から計算を再開できる必要がある．これは，計算ノードが多い上，通信路にインターネットを含むため，計算途中に一部の計算ノードとの通信に障害が起こる確率が高いためである．

ヘテロな環境への対応: 計算ノードの能力に応じてタスクを配分する仕掛けが必要である．各拠点の計算ノードおよび通信路の能力が異なる可能性が高く，能力の低い計算ノードがシステム全体のボトルネックになる可能性が高いためである．

NAT への対応: PC クラスタ内部のプライベート IP をもつ計算ノードを，ユーザー端末から利用できる必要がある．

3.3 NMR 蛋白質立体構造決定のためのグリッド向け GA システムの設計

3.3.1 グリッド上での並列化の基本方針

Ono らの手法において，最も計算時間を費やしている部分は，子の評価である．2.2 節のアルゴリズムのステップ 4「子の評価」において，各子個体の評価は全く独立に実行することができる．そこで，各子個体の評価を複数のワーカーで並列に実行することを考える．さらに，複数選択から世代交代までの世代ループを同時に複数実行することを考える．これにより，最大「1 世代あたりの子の生成数」×「世代ループ数」

☆ GA では，子個体が多少失われても，探索結果にはほとんど影響がないと考えられる．そのため，障害が起きた計算ノード上で評価されていた子個体を，再評価のために別の正常な計算ノードへ転送する機構は必須ではないと考えられる．

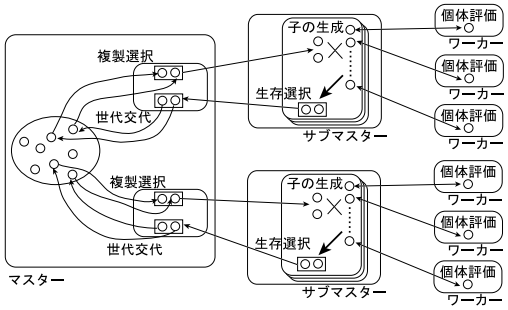

図 3 マスターとサブマスターの役割
Fig. 3 Roles of master and submasters

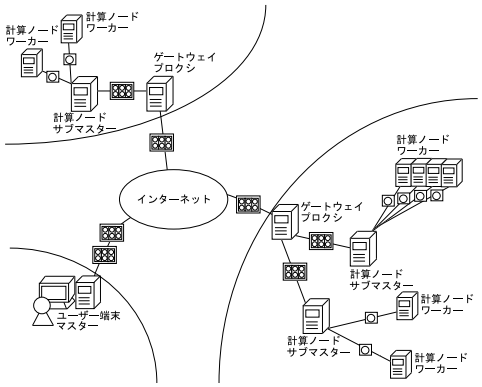

図 4 マスター，サブマスター，プロクシのノードへの割り付け
Fig. 4 Assignment of master, submaster and proxy to nodes

倍の高速化が可能であると考えられる．

本システムは，1,000 ワーカー程度のスケーラビリティが要求されている．1,000 ワーカーへ絶え間なく子を供給するため世代ループを複数実行した上で，全ワーカーとの通信を 1 つのマスターで行うことは困難であると考えられる．そこで，図 3 に示すように，本来マスターの機能を，マスターと複数のサブマスターに分担させる．マスターに複製選択，世代交代の機能を，サブマスターに子の生成，生存選択，ワーカーとの通信の機能を割り当てる．

インターネットを介した暗号化を伴う通信は，非常に通信遅延が大きくなる．そのため，通信データサイズおよび通信回数を抑えることが望ましい．そこで，図 4 に示すように，マスターをユーザー端末上，サブマスターを各サイトの PC クラスタ内の計算ノード上に配置し，マスターとサブマスターの間で，一度に複数世代分の親ペア，子ペアをやり取りさせる．

サブマスターを PC クラスタ内の計算ノード上に配置すると，計算ノードがプライベート IP を持つ場合，マスターからサブマスターへ直接通信ができない．そこで，図 4 に示すように，ゲートウェイ上に，通信の仲介をするプロクシを動作させる．

3.3.2 マスターの実装

マスターは，ユーザー端末上で動作するプログラムである．マスターの構成を図 5 に示す．マスター上では，メインスレッド，複製選択/世代交代スレッド，Ninf-G クライアントスレッドが協調動作している．

メインスレッドは，初期集団の生成，親ペア・キューの生成，複製選択/世代交代スレッドおよび Ninf-G クライアントスレッドの初期化を行う．複製選択/世代交代スレッドは，複製選択された親ペアを親ペア・キューへ登録し，Ninf-G クライアントスレッドから戻された子ペアを集団中の元の親ペアと置き換える．親ペア・キューに絶え間なく親ペアを供給するため，複数の複製選択/世代交代スレッドが動作している．Ninf-G クライアントスレッドは，プロクシと一対一に対応し，プロクシの初期化，プロクシへの親ペアの送信，プロクシからの子ペアの受信を行う．Ninf-G クライアントスレッドは，PC クラスタの計算能力に応じた数の親ペアを親ペア・キューから取り出し，一度にまとめてプロクシへ転送する．本論文では，一度にまとめて送られる複数の親ペアを「親ペア集合」☆ とよぶ．その後，Ninf-G クライアントスレッドは，親ペア集合に子の生成/生存選択を適用した結果である子ペア集合をプロクシから受け取り，子ペアをそれぞれ対応する複製選択/世代交代スレッドに戻す．

マスターとプロクシ間の通信はセキュリティが要求されることから，Ninf-G クライアントスレッドは Ninf-G[4),9)] を利用して実装されている．Ninf-G は，グリッドにおける遠隔手続き呼び出しによるプログラミングモデルのひとつである GridRPC によるプログラムの開発・実行を支援するミドルウェアである．Ninf-G は Globus Toolkit Version 2 (GTK2)[1)] および CoG[3)] を用いて実装されている．

何らかの障害によりプロクシとの通信が途絶えた場合，Ninf-G クライアントスレッドは，子ペアが戻らないことを複製選択/世代交代スレッドに通知した後，

☆ 親ペア集合の大きさを $|P|$，1 つの親ペアから生成される子個体数を n_{kid}，親ペア集合から生成される総子個体数を N_{kid} とする．本論文では，もし N_{kid} を n_{kid} で割った余りが 0 の場合は $|P| = N_{\mathrm{kid}}/n_{\mathrm{kid}}$，余りが $x(\neq 0)$ の場合は $|P| = N_{\mathrm{kid}}/n_{\mathrm{kid}} + 1$ と設定している．余りが $x(\neq 0)$ の場合，$|P|$ 個目の親ペアからは x 個の子が生成される．また，$|P|$ 個目の親ペアは親ペア・キューから削除されずに，別の Ninf-G クライアントスレッドによって処理される．したがって，1 つの親ペアは，複数の Ninf-G クライアントスレッドによって分割されて処理されることもある．この場合，複製選択/世代交代スレッドには複数の子ペアが戻ってくるため，それらの中から最良 2 個体を選択して集団中の元の親ペアと置き換える．

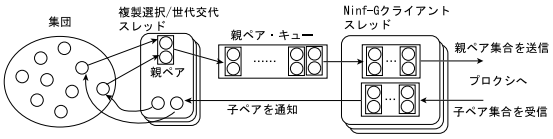

図 5 マスターの構成
Fig. 5 Architecture of master

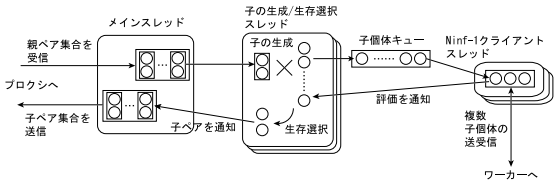

図 6 サブマスターの構成
Fig. 6 Architecture of submaster

終了する．これにより，全体の計算を止めずに，障害が起こったサイトのみを安全に切り離すことができる．

マスターは，Java を用いて実装されている．これは，強力なスレッド機構，メモリ管理機構および例外処理機構を利用できることから耐障害性を満たす実装を容易に構築できると考えられるためである．

3.3.3 プロクシの実装

プロクシは，ゲートウェイ上で動作する C で実装されたプログラムである．プロクシは，Ninf-G サーバーとしてマスターの Ninf-G クライアントスレッドにより起動される．その後，プロクシは，rsh を用いて計算ノード上にサブマスターを起動し，リダイレクトされた標準入出力を介して，マスターから送られてきた親ペア集合をサブマスターへ転送し，サブマスターから送られてきた子ペア集合をマスターへ返送する．

3.3.4 サブマスターの実装

サブマスターは，計算ノード上で動作するプログラムである．サブマスターの構成を図 6 に示す．サブマスターは，プロクシから rsh により起動される Java プログラムである．図 6 に示すように，サブマスターは，メインスレッド，子の生成/生存選択スレッド，Ninf-1 クライアントスレッドが協調動作している．

メインスレッドは，子個体キューの生成，子の生成/生存選択スレッドおよび Ninf-1 クライアントスレッドの初期化，標準入出力を介した親ペア集合/子ペア集合のプロクシとの送受信を行う．子の生成/生存選択スレッドは，プロクシから送られてきた親ペア集合から親ペアを１つずつ取り出した後，交叉・突然変異の適用により複数個の子を生成して，子個体キューへ登録する．その後，全ての子個体の評価結果が，Ninf-1 クライアントスレッドを介してワーカーから戻されたら，最良２個体を選択し，親ペア集合中の元の親ペアと入れ替える．子個体キューに絶え間なく子個体を供給するため，複数の子の生成/生存選択スレッドが動作している．Ninf-1 クライアントスレッドは，ワーカーの初期化，ワーカーへの子個体の送信，ワーカーからの子個体の受信を行う．Ninf-1 クライアントスレッドは，計算ノードの計算能力に応じた数の子個体を一度にまとめてワーカーへ転送する．その後，Ninf-1 クライアントスレッドは，ワーカーから返送されてきた子個体の評価結果を，対応する子の生成/生存選択スレッドへ戻す．

サブマスターとワーカー間の通信はセキュリティが要求されないため，Ninf-1 クライアントスレッドは高速な Ninf-1[4),10)] を利用して実装されている．Ninf-1 は，RPC による並列実行を支援するミドルウェアである．

何らかの障害によりワーカーとの通信が途絶えた場合，Ninf-1 クライアントスレッドは，子個体が戻らないことを子の生成/生存選択スレッドに通知した後，終了する．これにより，全体の計算を止めずに，障害が起こったワーカーのみを安全に切り離すことができる．

3.3.5 ワーカーの実装

ワーカーは，PC クラスタ内の計算ノード上で動作する C++で実装されたプログラムである．ワーカーは，サブマスターの Ninf-1 クライアントスレッドにより，Ninf-1 サーバーとして起動される．その際，ワーカーは，個体の評価に必要なアミノ酸の立体構造ライブラリなどのデータファイルを読み込む．その後，ワーカーは，サブマスターの Ninf-1 クライアントスレッドから子個体を複数個まとめて受け取り，全ての子個体を評価した後，それらをサブマスターの Ninf-1 クライアントスレッドへ送り返す，という動作を繰り返す．

3.3.6 スライド転送方式による通信遅延の隠蔽

マスターとサブマスター間は，暗号/復号化を伴うインターネット上の通信区間を含むため，非常に通信のオーバーヘッドが大きい．図 7(a) に示すように，通常の RPC の場合，マスターは，親ペア集合 P_i をサブマスターへ送った後，戻り値である子ペア集合 C_i が帰ってくるまで，P_{i+1} を送信しない．そのため，サブマスターが C_i を送り返してから P_{i+1} を受け取るまでの間，PC クラスタ上の全ワーカーが遊んでしまう．

本論文では，ワーカーの遊び時間を見かけ上なくす工夫として，図 7(b) に示すスライド転送方式を導入する．スライド転送方式におけるマスター，サブマスターの動作のシナリオを以下に示す：

表 1 各サイトの PC クラスタのスペック
Table 1 Spec of PC clusters at each site

サイト	CPU	プロセッサ数	ソフトウェア
徳島大小野研究室	Athlon MP 2000+	126	Globus : 2.4.*
	Athlon MP 2800+	84	Ninf-G : 2.2.0
東工大合田研究室	Pentium Ⅲ 1.4GHz	70	Java : 1.4.*
東京電機大藤沢研究室	Athlon MP 2400+	78	gcc : 2.95.*
	Opteron 240 (64bit)	30	glibc : 2.2.*
東工大松岡研究室	Athlon MP 1900+	194	CA : ApGrid
	Athlon MP 2000+	66	
	Opteron 242 (32bit)	74	
産総研グリッドセンター	Xeon 3.06GHz	374	

表 2 1 個体あたりの平均評価時間と Pentium III 1.4GHz の性能を 1.0 としたときの各 CPU の性能比
Table 2 Average evaluation time per an individual and the performance ratio of each CPU when the performance of Pentium III 1.4GHz is 1.0

サイト	CPU	平均評価時間 [ms]	性能比
徳島大小野研究室	Athlon MP 2000+	2445.7745	1.1950
	Athlon MP 2800+	1761.8762	1.6588
東工大合田研究室	Pentium Ⅲ 1.4GHz	2922.6529	1.0000
東京電機大藤沢研究室	Athlon MP 2400+	2178.7201	1.3415
	Opteron 240 (64bit)	2192.8993	1.3328
東工大松岡研究室	Athlon MP 1900+	2490.8498	1.1734
	Athlon MP 2000+	2413.2505	1.2111
	Opteron 242 (32bit)	2506.4472	1.1661
産総研グリッドセンター	Xeon 3.06GHz	1306.4318	2.2371

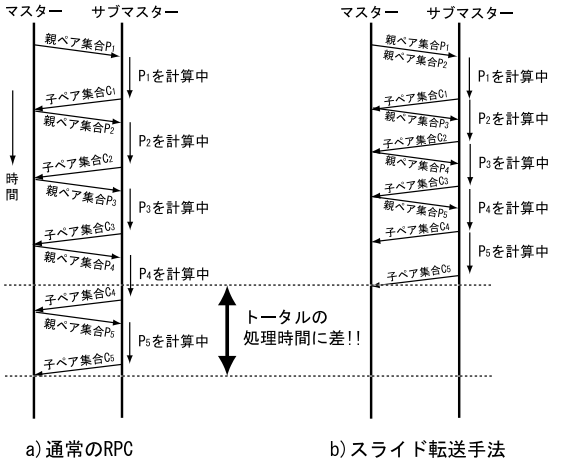

図 7 通常の RPC とスライド転送手法の比較
Fig. 7 Comparison of normal RPC and slide transfer method

(1) あらかじめ P_1 と P_2 がマスターからサブマスターへ転送される．$i = 1$ に初期化する．
(2) サブマスターが P_i の処理を開始する．
(3) P_{i+2} がマスターからサブマスターへ転送される．RPC の戻り値として C_i がサブマスターからマスターへ戻される．
(4) $i = i + 1$ として，ステップ 2 へ．

これにより，サブマスターからマスターへ C_i が送り返されから，サブマスターがマスターから P_{i+2} を受け取るまでの間においても，サブマスターは P_{i+1} の処理をできる．したがって，ワーカーの遊び時間を見かけ上なくすことができると考えられる．

4. 実　　験

本節では，提案システムのスケーラビリティとスライド転送手法の有効性，提案システムの耐障害性を検証する．さらに，実運用を想定した運用試験も行う．

実験に用いたグリッドテストベッドは，徳島大小野研 (Site 1)，東工大合田研 (Site 2)，東京電機大藤沢研 (Site 3)，東工大松岡研 (Site 4)，産総研グリッドセンター (Site 5) の 5 サイトの PC クラスタをインターネットで相互接続した合計 1,196CPU から構成されるグリッド計算環境である．各サイトの PC クラスタのスペックを表 1 に示す．

対象問題は，78 残基のアミノ酸からなる hmg2b 蛋白質の立体構造決定問題を用いた．

4.1 実験１：スケーラビリティおよびスライド転送手法の有効性の検証

本実験で用いるグリッドテストベッドは，能力の異なる CPU を搭載した PC クラスタで構成されている．そこで，まず，スケーラビリティを調べるための予備実験として，各 CPU の性能比を調べる実験を行った．本予備実験では，各 CPU を用いて，11,100 個の個体を評価した．表 2 に，1 個体あたりの平均計算時間，

表 3 Pentium III 1.4GHz の能力を 1.0 としたときの正規化 CPU 数

サイト	実 CPU 数	正規化 CPU 数
小野研	210	289.9111
小野研+合田研	280	359.9111
小野研+合田研+藤沢研	388	504.5279
小野研+合田研+藤沢研+松岡研	822	1014.9840
小野研+合田研+藤沢研 +松岡研+産総研	1196	1851.6692

および，Pentium III 1.4GHz の性能を 1.0 としたときの各 CPU の性能比を表している．また，表 2 に基づき，CPU 数を Pentium III 1.4GHz で正規化した「正規化 CPU 数」を表 3 に示す．

次に，提案システムのスケーラビリティおよびスライド転送手法の有効性を調べる実験を行った．本実験では，Site 1〜5 の順にサイト数を増やしていったとき，500 世代を実行するのに要した時間を計測した．集団サイズを 500，1 世代あたりの生成子個体数を 200，複製選択/世代交代スレッドの数を 10(サイト数 1)，14(サイト数 2), 16(サイト数 3), 36(サイト数 4), 50(サイト数 5)，子の生成/世代交代スレッドの数を 18 とした．各サイトで生成される子個体数が CPU 数× 4 と同じになるように，マスターが各サイトに一度に投げる親ペア数を設定した．サブマスターが各ワーカーに一度に投げる子個体数を 1 に設定した．本問題における 1 個体あたりのデータ量は，約 11.5KB である．スライド転送方式を用いた場合と用いなかった場合の結果を図 8，図 9 に示す．図 8 は，正規化 CPU 数と計算時間の関係のグラフを示す．図 9 は，正規化 CPU 数と Pentimum III 1.4GHz の計算速度を 1.0 としたときのスピードアップ比の関係のグラフである．グラフは，いずれも独立 3 試行の平均値である．図 9 より，スライド転送手法を用いた提案システムが理想に近いリニアなスケーラビリティを示していることがわかる．また，スライド転送手法を用いた場合と用いなかった場合を比較することにより，スライド転送手法の有効性を確認できる．

4.2 実験２：耐障害性の検証

以下の状況を人為的に発生させた結果，システム全体の処理は止まらずに正常に計算を続けることを確認した．

- マスター⇔プロクシ，プロクシ⇔サブマスター，サブマスター⇔ワーカーのコネクションを切断
- マスター⇔プロクシ，プロクシ⇔サブマスター，サブマスター⇔ワーカーの通信路を物理的に切断
- プロクシ，サブマスター，ワーカーのプログラムを異常終了

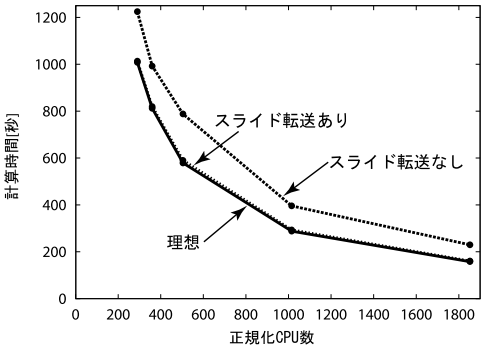

図 8 正規化 CPU 数 vs. 計算時間 (78 残基)
Fig. 8 Normalized No. of CPUs vs. Computation time (78 residues)

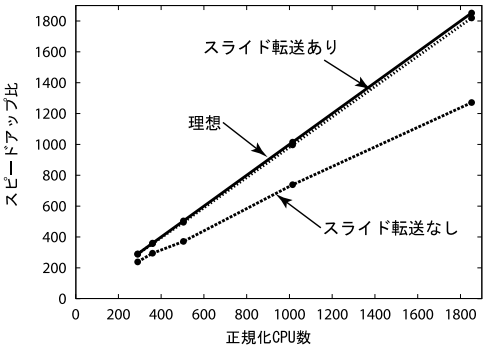

図 9 正規化 CPU 数 vs. スピードアップ比 (78 残基)
Fig. 9 Normalized No. of CPUs vs. Speed up (78 residues)

また，マスターが異常終了した場合でも，途中から計算を再開できることを確認した．

4.3 実験３：実運用を想定した運用試験

実運用を想定して，サイト数を 5，CPU 数を 1,196，打ち切り世代を 30,000 世代として実験を行った．その他のパラメータは実験 1 と同じである．その結果，約 2 時間 40 分 (3 試行の平均) で正常に終了することを確認した．同じ計算を Pentium III 1.4GHz のシングル CPU マシンで実行すると約 200 日かかる．これより，提案システムにより，実際の規模の問題でも，現実的な時間内に計算を終了することが出来ることが確認された．

5. 考　　察

本節では，残基数を 49 に減らしたときに，提案システムの性能がどのように変化するかについて，実験的に考察する．残基数 49 の問題に提案システムを適用した際の，正規化 CPU 数と Pentimum III 1.4GHz の計算速度を 1.0 としたときのスピードアップ比の関係

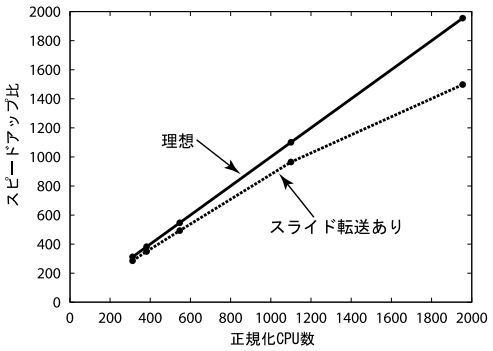

図 10 正規化 CPU 数 vs. スピードアップ比 (49 残基)
Fig. 10 Normalized No. of CPUs vs. Speed up (49 residues)

のグラフを図 10 ☆ に示す．これより，4 サイトまでは比較的良好なスケーラビリティを見せている．しかし，5 サイトまで増やすと，通信遅延を隠蔽しきれなくなり，理想的なスピードアップを示す直線から大きく離れてしまっている．ちなみに，Pentium III 1.4GHz のシングル CPU マシン上で，残基数 49 の 1 個体の平均評価時間は 1.003 秒であった．また，1 個体あたりのデータ量は 7.76KB であった．しかし，4 サイトを利用することで，30,000 世代実行するのにかかる時間は約 1 時間 45 分程度であり，実運用上，十分な性能であると考えられる．

6. おわりに

本論文では，NMR 蛋白質立体構造決定のためのグリッド向け GA システムを提案した．5 サイト/1,196CPU から構成されるグリッドテストベッドを用いて，提案システムのスケーラビリティ，耐障害性の評価を行った．その結果，78 残基の問題において，1,196CPU まで理想的なスケーラビリティを示すことを確認した．また，さまざまな障害が起こっても，止まらずに計算を継続することを確認した．さらに，最適化計算が終了するまで実行したところ，Pentium III 1.4GHz では 200 日かかるのに対し，提案システムでは約 2 時間 40 分で正常終了することを確認した．

今後は，実際の蛋白質への適用を通じて，交叉，突然変異などの探索オペレータの改良をしたいと考えている．NMR 蛋白質立体構造決定の専門家が本システムを利用しやすいように，WEB ポータルインターフェースの開発も今後の課題である．また，本論文で提案したグリッド上での並列化手法は，広範囲の GA に適用できると考えられる．今後は，本並列化手法をフレームワーク化し，さまざまな GA に適用したいと考えている．

謝辞 本研究を遂行するに当たって，ご協力いただいた東京工業大学の田中氏をはじめとする松岡研究室の皆様，東京工業大学の合田助教授をはじめとする合田研究室の皆様，東京電機大学の藤沢助教授をはじめとする藤沢研究室の皆様，産業技術総合研究所の田中氏をはじめとするグリッドセンターの皆様，東京工業大学の小島教授，徳島大学の松原君，中井君，多畑君に感謝の意を表します．

☆ 本グラフは，残基数 49 の問題で 1 個体あたりの平均評価時間を再計測し，その結果を用いて再計算された正規化 CPU 数を用いている．そのため，グラフの横軸の範囲が残基数 78 の場合と微妙に変化している．

参 考 文 献

1) Foster, I., Kasseleman C.: Globus: A metacomputing infrastructure toolkit, Int'l Journal of Supercomputing Applications, 11(2), pp. 115–128 (1997).
2) Goldberg, D. E.: *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley (1989).
3) Laszewski, G., Foster, I., Gawor, J., Smith, W., Tuecke, S.: CoG Kits: A Bridge between Commodity Distributed Computing and High-Performance Grids, In ACM 2000 Java Grande Conf. pp. 97–106 (2000).
4) http://ninf.apgrid.org/welcome.shtml
5) Ono, I. Fujiki, H., Ootsuka, M., Nakashima, N., Ono, N. and Tate, S.: Global Optimization of Protein 3-Dimensional Structures in NMR by A Genetic Algorithm, Proc. 2002 Congress on Evolutionary Computation, pp. 303–308 (2002).
6) 伊藤隆：効率的で迅速な NMR タンパク質構造解析法の模索, 実験医学, Vol. 19, No. 8, pp. 954–957 (2001).
7) 喜多一，山村雅幸：機能分担仮説に基づく GA の設計指針, 計測と制御, Vol. 38, No. 10, pp. 612–617 (1999).
8) 佐藤浩, 小野功, 小林重信：遺伝的アルゴリズムにおける世代交代モデルの提案と評価, 人工知能学会誌, Vol. 12, No. 5, pp. 734–744 (1997).
9) 田中良夫, 中田秀基, 平野基孝, 佐藤三久, 関口智嗣：Globus による Grid RPC システムの実装と評価, 情報処理学会ハイパフォーマンスコンピューティング研究会, Vol. 2001, No. 77, pp. 165–170 (2001).
10) 中田秀基, 高木浩光, 松岡 聡, 長嶋雲兵, 佐藤 三久, 関口 智嗣：Ninf による広域分散並列計算, 情報処理学会論文誌 Vol. 39, No. 6, pp. 1818–1826 (1998).