

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^{†2} 中島 直 敏^{†1}
 小野 典 彦^{†2} 中田 秀 基^{†1,†3} 松岡 聡^{†1,†4}
 関口 智 嗣^{†3} 楯 真 一^{†5}

本論文では、Ono らが提案した NMR 蛋白質立体構造決定のための遺伝アルゴリズム (GA) を、複数の WAN 上の PC クラスタ群から構成されるグリッド上で並列化したシステムを提案し、提案システムの性能評価を行った結果を報告する。提案システムは、マスタ、サブマスタ、ワーカから構成される階層的なマスタ・ワーカ方式を用いて並列化されている。マスタと各 PC クラスタ間の通信はセキュアな GridRPC ミドルウェア Ninf-G を用いて、また、PC クラスタ内の通信は高速な GridRPC ミドルウェア Ninf-1 を用いて実現されている。さらに、提案システムでは、Ninf-G によるインターネット上の通信遅延を隠蔽するため、スライド転送手法を導入している。5 サイト/1,196 CPU から構成されるグリッドテストベッドで、78 残基からなる蛋白質の立体構造決定問題を用いて、提案システムの性能評価を行った結果、高い並列化効率を示すことが確認された。

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

ISAO ONO,^{†1} NAOAKI MIZUGUCHI,^{†2} NAOTOSHI NAKASHIMA,^{†1}
 NORIHIKO ONO,^{†2} HIDEMOTO NAKADA,^{†1,†3} SATOSHI MATSUOKA,^{†1,†4}
 SATOSHI SEKIGUCHI^{†3} and SHIN-ICHI TATE^{†5}

In this paper, we parallelize the genetic algorithm (GA) for NMR protein three-dimensional structure determination, which has been proposed by Ono *et al.*, on a grid that consists of multiple PC clusters on the WAN and report some results on the performance evaluation of the proposed system. The proposed system is parallelized with the hierarchical master-worker paradigm and consists of a master, submasters and workers. The communication between the master and each PC cluster is realized with Ninf-G, which is a secure GridRPC middleware, and that in each PC cluster is implemented by using Ninf-1, which is a fast GridRPC middleware. In the proposed system, we employ the *slide transfer technique* in order to hide the latency of communication on the Internet by using Ninf-G. The experimental results on the grid testbed consisting of 5 sites/1,196 CPUs showed that the proposed system effectively utilized computing resources on the grid testbed when it was applied to a problem of determining the three-dimensional structure of a 78-residue protein.

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 1 つであり、核磁気共鳴法 (NMR) は有望な構造解析技術の 1 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 ¹H 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

†1 東京工業大学

Tokyo Institute of Technology

†2 徳島大学

The University of Tokushima

†3 産業技術総合研究所

National Institute of Advanced Industrial Science and Technology

†4 国立情報学研究所

National Institute of Informatics

†5 生物分子工学研究所

Biomolecular Engineering Research Institute

め、専門家でも1つの蛋白質の立体構造決定に数カ月を要する。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⁸⁾。

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⁶⁾。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200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、解候補を集団として保持しながら探索を行う最適手法である。GAは、現在の集団の情報を用いて新たな集団を生成し、その2つの集団中から評価値の良い解候補を選択して、次の集団を形成するというプロセスを繰り返しながら探索を行う。ここで、新たに生成された集団中の解候補の評価計算は互いに独立に実行できるため、GAは本質的に並列化が可能である。Onoらの手法⁶⁾において、計算時間の大半を占めているのは解候補である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クラスターなどの計算資源を、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クラスター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^{5),15)} および Ninf-G^{5),13)} を用いて Ono らの GA⁶⁾ 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、東工大合田研、東京電機大藤沢研、徳島大小野研の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

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、蛋白質中に多数含まれる水素原子核¹Hに由来する化学シフトとよばれるシグナルデータを得ることができる。化学シフトは、各¹Hを取り巻く電子密度に依存することから、各¹Hに固有の値である。いくつかの条件で観測されたシグナルデータを用いることにより、観測された化学シフトを元の¹Hに帰属していくことが可能である。この作業は、連鎖帰属とよばれ、現状で1~2週間程度の作業である。

連鎖帰属に用いたものとは別のある特別な測定条件において、NOE (Nuclear Overhauser Effect) シグナルとよばれる化学シフトのペアを数多く測定することができる。NOEシグナルは、お互いの距離が十分に近い(5Å程度)¹Hのペアから観測されることから、連鎖帰属で帰属した化学シフトを手がかりに、NOEシグナルを対応する¹Hペアに帰属することにより、¹H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¹H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⁶⁾。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8Å以下の¹Hペアを列挙したものであり、観測NOEをすべ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どうし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)³⁾ 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

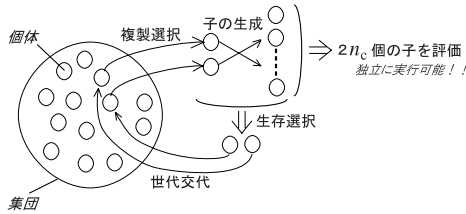


図 1 Minimal Generation Gap (MGG) モデル
Fig. 1 Minimal Generation Gap (MGG) model.

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¹⁰⁾。以下、Ono らの手法⁶⁾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、 $-1.0^\circ \sim +1.0^\circ$ の一様乱数を加える。

世代交代モデルは、Minimal Generation Gap (MGG)¹¹⁾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。
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で生成されたすべての子個体の評価値を計算する。
5. 生存選択： 両親と生成されたすべての子を含め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。

上記、ステップ4「子の評価」において、各子個体の評価はまったく独立に実行することができることに注意されたい。

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。ユーザ端末

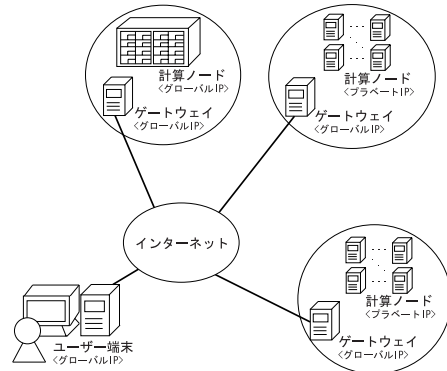


図 2 本論文で想定するグリッド計算環境
Fig. 2 Grid computation environment assumed in this paper.

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 を持ち、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： 1,000 CPU 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。

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、計算ノードが多いうえ、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

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。

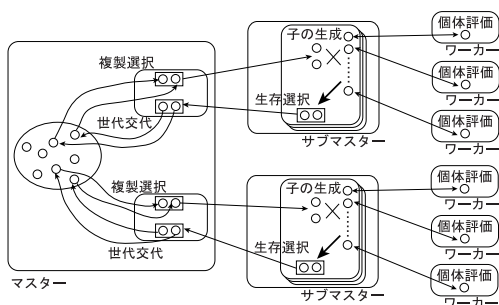


図 3 マスタとサブマスタの役割

Fig. 3 Roles of master and submasters.

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 を持つ計算ノードを，ユーザ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，各子個体の評価はまった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で並列に実行することを考える．さらに，複製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」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 ワーカ程度のスケーラビリティが要求されている．1,000 ワーカ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うえで，全ワーカとの通信を 1 つのマスタ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，本来マスタの機能を，マスタと複数のサブマスタに分担させる．マスタに複製選択，世代交代の機能を，サブマスタに子の生成，生存選択，ワーカとの通信の機能を割り当てる．

マスタとワーカを直接通信させる単純なマスタ・ワーカ方式で Ono らの GA を並列化し，100 個の Athlon MP 2800+ から構成される PC クラスタを用いて，78 残基蛋白質立体構造決定問題への適用を行ったところ，マスタ (Dual Athlon MP 2800+ の PC) の負荷率は 25% 程度であった．本実験において通信データの暗号化は行っていない．したがって，1 つのマスタが，通信を暗号化したうえで，1,000 ワーカと直接通信を行うことは困難と考えられる．

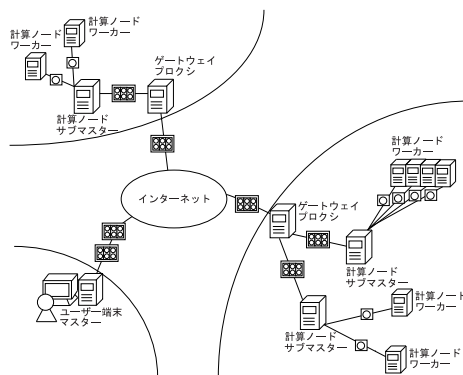


図 4 マスタ，サブマスタ，プロキシのノードへの割付け

Fig. 4 Assignment of master, submaster and proxy to nodes.

インターネットを介した暗号化をとまなう通信は，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，マスタをユーザ端末上，サブマスタを各サイトの PC クラスタ内の計算ノード上に配置し，マスタとサブマスタの間で，一度に複数世代分の親ペア，子ペアをやりとりさせる．

サブマスタを 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，マスタからサブマスタ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キシを動作させる．

本論文では，GridRPC ミドルウェアである Ninf-G^{5),13)} および Ninf-1^{5),15)} を用いて，提案システムをグリッド上に実装する．GridRPC は，グリッド上でクライアント・サーバ型の遠隔手続呼び出しを実現するためのプログラミングモデルであり，GridRPC ミドルウェアを用いることにより，グリッド上でのマスタ・ワーカ方式のプログラムが容易に実装可能である．GridRPC の実装には，Ninf-G，Ninf-1 のほかに，Net-Solve¹⁾ などが提案されている．Ninf-G は，Globus Toolkit²⁾ および CoG⁴⁾ を利用した GridRPC ミドルウェアであり，強力なユーザ認証機能と通信路の暗号化機能を備えたものとなっている．一方，Ninf-1 は，Ninf-G のような強力なユーザ認証機能および通信路の暗号化機能を備えないかわりに，非常に高速な通信を実現している．そこで，本論文では，マスタとプロキシ間のインターネット上の通信は，セキュリティを確保する必要があることから，Ninf-G を用いることにする．また，サブマスタとワーカ間の PC クラスタ内の通信は，セキュリティを考慮する必要がないと考えられることから，Ninf-1 を用いる．これにより，PC

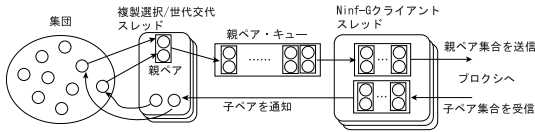


図5 マスタの構成
Fig.5 Architecture of master.

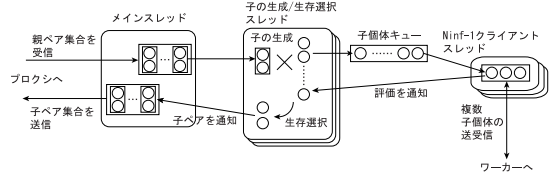


図6 サブマスタの構成
Fig.6 Architecture of submaster.

クラスタ内の通信遅延を低減することができると考えられる。プロクシとサブマスタ間の通信は、PC クラスタ内の通信であり、通信を仲介すればよいだけなので、rsh を用いて実装することにする。

3.3.2 マスタの実装

マスタは、ユーザ端末上で動作するプログラムである。マスタの構成を図5に示す。マスタ上では、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1対1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^{5),13)} を利用して実装されている。

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、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。

メインスレッドは、定期的に集団の情報をファイルに保存している。そのため、何らかの障害によりマスタが止まってしまっても、一番最近に保存された集団情報をファイルから読み込むことにより、そこから探索を再開することができる。

マスタは、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 サーバとしてマスタの 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、マスタから送られてきた親ペア集合をサブマスタへ転送し、サブマスタから送られてきた子ペア集合をマスタへ返送する。

3.3.4 サブマスタの実装

サブマスタは、計算ノード上で動作するプログラムである。サブマスタの構成を図6に示す。サブマスタは、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

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 $|P| = N_{kid}/n_{kid}$ 、余りが $x(x \neq 0)$ の場合は $|P| = N_{kid}/n_{kid} + 1$ と設定している。余りが $x(x \neq 0)$ 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。

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、すべ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^{5),15)} を利用して実装されている。

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、サブマスタの Ninf-1 クライアントスレッドにより、Ninf-1 サーバとして起動される。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、すべての子個体を評価した後、それらをサブマスタの 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

マスタとサブマスタ間は、暗号/復号化をともな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、マスタは、親ペア集合 P_i をサブマスタへ送った後、戻り値である子ペア集合 C_i が返ってくるまで、 P_{i+1} を送信しない。そのため、サブマスタが C_i を送り返してから P_{i+1} を受け取るまでの

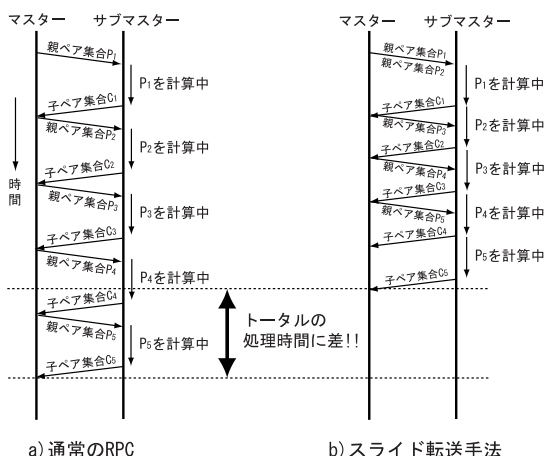


図 7 通常の RPC とスライド転送手法の比較

Fig. 7 Comparison of normal RPC and slide transfer method.

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、サブマスタの動作のシナリオを以下に示す：

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

これにより、サブマスタからマスタへ C_i が送り返されから、サブマスタがマスタから P_{i+2} を受け取るまでの間においても、サブマスタは 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

本章では、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、東工大合田研、東京電機大藤沢研、東工大松岡研、産総研グリッドセンターの 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 1,196 CPU 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。

プログラム実行に必要なデータファイルやプログラム本体は、あらかじめ想定されるすべてのノードに配布されている必要がある。この作業は非常に煩雑になるため、提案システムにおいては、この作業を自動化するシェルスクリプトを自動的に生成するプログラムを提供している。本プログラムは、ファイル名や配布先のノード名などの情報を定義したファイルを引数として渡すことにより動作する。

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	Athlon MP 2000+	126	Globus : 2.4.*
	Athlon MP 2800+	84	Ninf-G : 2.2.0
東工大合田研	Pentium III 1.4 GHz	70	Java : 1.4.*
東京電機大藤沢研	Athlon MP 2400+	78	gcc : 2.95.*
	Opteron 240 (64 bit)	30	glibc : 2.2.*
東工大松岡研	Athlon MP 1900+	194	CA : ApGrid
	Athlon MP 2000+	66	
	Opteron 242 (32 bit)	74	
産総研グリッドセンター	Xeon 3.06 GHz	374	

表 2 1 個体あたりの平均評価時間と Pentium III 1.4 GHz 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 1.4 GHz is 1.0.

サイト	CPU	平均評価時間 [ms]	性能比
徳島大小野研	Athlon MP 2000+	2,445.7745	1.1950
	Athlon MP 2800+	1,761.8762	1.6588
東工大合田研	Pentium III 1.4 GHz	2,922.6529	1.0000
東京電機大藤沢研	Athlon MP 2400+	2,178.7201	1.3415
	Opteron 240 (64 bit)	2,192.8993	1.3328
東工大松岡研	Athlon MP 1900+	2,490.8498	1.1734
	Athlon MP 2000+	2,413.2505	1.2111
	Opteron 242 (32 bit)	2,506.4472	1.1661
産総研グリッドセンター	Xeon 3.06 GHz	1,306.4318	2.2371

4.1 実験 1：スケ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、および、Pentium III 1.4 GHz 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 1.4 GHz 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、徳島大小野研、徳島大小野研 + 東工大合田研、徳島大小野研 + 東工大合田研 + 東京電機大藤沢研、徳島大小野研 + 東工大合田研 + 東京電機大藤沢研 + 東工大松岡研、徳島大小野研 + 東工大合田研 + 東京電機大藤沢研 + 東工大松岡研 + 産総研グリッドセンターと利用するサイトを順に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。マスタの管理している集団の集団サイズは 500 とした。マスタにおける複製選択/世代交代スレッドの同時実行数は、徳島大小野研の PC クラスタのみを利用する場合に 10、徳島大小野研 + 東工大合田研の PC クラスタを利用す

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

Table 3 The normalized number of CPUs when the performance of Pentium III 1.4. GHz is 1.0.

利用サイト	実 CPU 数	正規化 CPU 数
徳島大小野研	210	289.9111
徳島大小野研 + 東工大合田研	280	359.9111
徳島大小野研 + 東工大合田研 + 東京電機大藤沢研	388	504.5279
徳島大小野研 + 東工大合田研 + 東京電機大藤沢研 + 東工大松岡研	822	1,014.9840
徳島大小野研 + 東工大合田研 + 東京電機大藤沢研 + 東工大松岡研 + 産総研グリッドセンター	1,196	1,851.6692

る場合に 14、徳島大小野研 + 東工大合田研 + 東京電機大藤沢研の PC クラスタを利用する場合に 16、徳島大小野研 + 東工大合田研 + 東京電機大藤沢研 + 東工大松岡研の PC クラスタを利用する場合に 36、徳島大小野研 + 東工大合田研 + 東京電機大藤沢研 + 東工大松岡研 + 産総研グリッドセンターの PC クラスタを利用する場合に 50 と設定した。マスタが各サイトに一度に投げる親ペア数は、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設定した。各サブマスタ上における子の生成/生存選択スレッドの同時実行数を 18、それぞれの子の生成/生存選択スレ

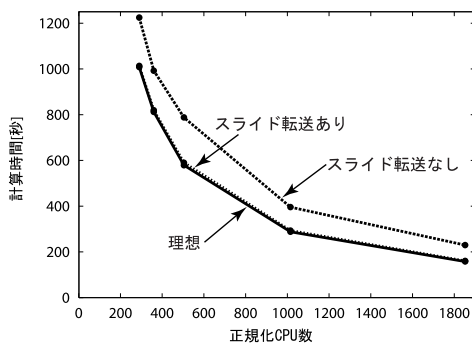


図 8 正規化 CPU 数 vs. 計算時間 (78 残基)

Fig. 8 Normalized no. of CPUs vs. computation time (78 residues).

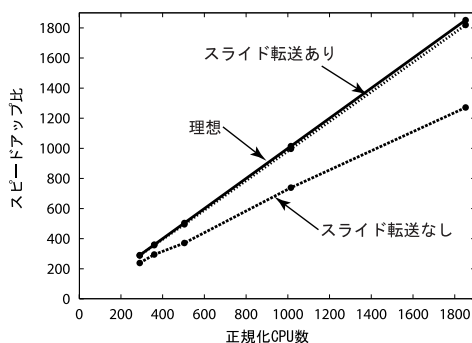


図 9 正規化 CPU 数 vs. スピードアップ比 (78 残基)

Fig. 9 Normalized no. of CPUs vs. speed up (78 residues).

ドにおける 1 世代あたりの生成子個体数を 200 と設定した。サブマスタが各ワーカーに一度に投げける子個体数は 1 と設定した。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 Pentium 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ニアなスケールビリティを示していることが分かる。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 ⇔ ブロキシ, ブロキシ ⇔ サブマスタ, サブマスタ ⇔ ワーカーのコネクションを切断
- マスタ ⇔ ブロキシ, ブロキシ ⇔ サブマスタ, サ

ブマスタ ⇔ ワーカーの通信路を物理的に切断

- ブロキシ, サブマスタ, ワーカーのプログラムを異常終了

また、マスタ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、打ち切り世代を 30,000 世代として実験を行った。実験には、徳島大小野研 + 東工大合田研 + 東京電機大藤沢研 + 東工大松岡研 + 産総研グリッドセンターのすべての PC クラスタを利用した。ワーカーとして利用した CPU 数は合計 1,196 である。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できることが確認された。

5. 考 察

5.1 残基数とスケールビリティに関する考察

前章で用いた 78 残基の hmg2b 蛋白質よりも残基数が少ない蛋白質を対象とした場合、1 個体あたりの評価時間が短くなるため、提案システムはスケールビリティが劣化すると考えられる。そこで、hmg2b 蛋白質を構成する 3 本の α -helix のうち 1 本を取り去ることにより人工的に作成した残基数 49 の問題を対象に、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 とした場合、1 個体の平均評価時間は 1.003 秒である。これは、前章で用いた残基数 78 の蛋白質の約 1/3 である。また、1 個体あたりのデータ量は 7.76 KB であった。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 Pentium 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ル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。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。

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。

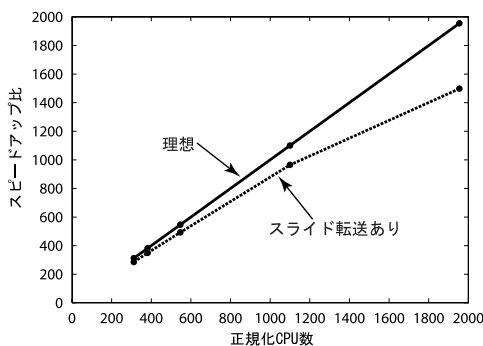


図 10 正規化 CPU 数 vs. スピードアップ比 (49 残基)

Fig. 10 Normalized no. of CPUs vs. speed up (49 residues).

5.2 並列化の探索性能への影響に関する考察

提案システムでは、並列化効率を向上させるために、複製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形で、オリジナルの MGG のアルゴリズムを変更している。そのため、探索性能が劣化してしまう恐れがある。ここで、探索性能とは、「得られた最良解の精度」と「最良解が得られるまでに評価した解候補の数」である。これに関して、著者らのグループでは、関数最適化や組合せ最適化における典型的なベンチマーク問題を用いて、系統的な実験を行っている⁹⁾。その結果、NMR 蛋白質立体構造決定問題のような変数の組合せを最適化する問題においては、世代ループの同時実行数の増加による探索性能の劣化はほとんど見られないという結果を得ている。探索性能の劣化が最も見られたのは、変数間に非常に強い依存関係を持つ関数最適化問題であった。この問題では、得られた最良解の精度は変わらず、最良解が得られるまでに評価した解候補の数が若干増加した。しかし、その増加は、世代ループの同時実行数を増やしたことにより得られる並列化効率の向上に比べれば十分に小さな値であった。詳細は文献 9) を参照されたい。

6. 関連研究

本研究の関連研究として、合田らによるグリッド上での分枝限定法の並列化に関する研究があげられる⁷⁾。合田らは、細粒度のタスクをグリッド上で効率良く実行するために、階層的マスタ・ワーカ方式を用いて並列化を行う方法を提案している。合田らのシステムでは、スーパーバイザ(本研究におけるマスタに相当)とマスタ(本研究におけるサブマスタに相当)間の通信に Ninf-G、マスタとワーカ(本研究におけるワーカに相当)間の通信に Ninf-1 を用いている。したがって、本研究における提案システムは、合田らの提案し

た階層的マスタ・ワーカ方式の GA への適用例として位置づけられると考えられる。しかし、合田らのシステムには、本研究で提案したスライド転送方式に相当する工夫が入っていないため、インターネット上の通信遅延を隠蔽することができていないという問題があると考えられる。

大規模アプリケーションをグリッド上で並列化するプログラミング方式としては、合田らの研究や本研究のような GridRPC をカスケードする方式のほかに、GridRPC と MPI を組み合わせる方式も提案されている¹²⁾。また、中田らは、グリッドの階層構造を考慮した階層型分散実行環境 Jojo¹⁶⁾ を提案し、そのうえで組合せ最適化問題を容易に解くための最適化システム jPop¹⁴⁾ を提案している。jPop では、すべてのプログラムが Java で書かれていることが前提であるため、jPop を用いるためには、個体を評価するためのワーカプログラムを C++ から Java へ書き換える必要があると考えられる。

7. 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/1,196 CPU 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、1,196 CPU 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 1.4 GHz 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。

今後の課題としては、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を改良すること、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 ポータルインタフェースを開発することなどが考えられる。また、本論文で提案したグリッド上での並列化手法において、NMR 蛋白質立体構造決定問題に依存している部分は、個体のデータサイズと個体の評価時間のみである。したがって、個体表現、交叉方法、評価器の部分を自由に差し替えられるようにフレームワーク化すれば、さまざまな問題へ適用可能であると考えられる。したがって、提案手法のフレームワーク化も今後の課題である。さらに、提案システムにおいては、ユーザは PC クラスタ内部の情報を詳細に知っている必要がある。すなわち、ユーザは各サイトの PC クラスタ内のノードの

スペックや各ノードが現在利用可能かどうかについて調べて、それらの情報をサブマスタへ渡す必要があるため、ユーザにとって煩雑である。サブマスタで自動的にそれらの情報を収集させる機能を実現し、ユーザがPC クラスタ内部の情報を知らなくてもシステムを利用できるようにすることも今後の課題である。

謝辞 本研究は、科学技術振興機構計算機科学技術活用型特定研究開発推進事業 (ACT-JST) 研究課題「コモディティグリッド技術によるテラスケール大規模数値最適化」の援助を受けています。本研究を遂行するにあたって、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。

参考文献

- 1) Casanova, H. and Dongarra, J.: Netsolve: A Network Server for Solving Computational Science Problems, *International Journal of Super-computer Applications and High Performance Computing*, Vol.1, No.3, pp.212-223 (1997).
- 2) Foster, I. and Kasseleman C.: Globus: A meta-computing infrastructure toolkit, *Int'l Journal of Supercomputing Applications*, Vol.11, No.2, pp.115-128 (1997).
- 3) Goldberg, D.E.: *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley (1989).
- 4) Laszewski, G., Foster, I., Gawor, J., Smith, W. and Tuecke, S.: CoG Kits: A Bridge between Commodity Distributed Computing and High-Performance Grids, *ACM 2000 Java Grande Conf.*, pp.97-106 (2000).
- 5) <http://ninf.apgrid.org/welcome.shtml>
- 6) 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).
- 7) 合田憲人, 大角知孝: グリッド上での並列分枝限定法アプリケーション, インターネットカンファレンス 2004 講演論文集, pp.73-81 (2004).
- 8) 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).
- 9) 今出広明, 水口尚亮, 小野 功, 小野典彦: 世代交代モデル MGG の並列化とその性能評価, 第 32 回知能システムシンポジウム資料, pp.163-168 (2005).
- 10) 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).
- 11) 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).
- 12) 武宮 博, 田中良夫, 中田秀基, 関口智嗣: MPI と GridRPC を利用した大規模 Grid アプリケーションの開発と実行: Hybrid QM/MD シミュレーション, SACSIS 2005 (先進的計算基盤システムシンポジウム) (2005) (to appear).
- 13) 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).
- 14) 中川伸吾, 中田秀基, 松岡 聡: 並列組合せ最適化システム jPoP の分枝限定法の実装, 第 16 回コンピュータシステム・シンポジウム予稿集 (2004).
- 15) 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).
- 16) 中田秀基, 松岡 聡, 関口智嗣: Java による階層型グリッド環境 Jojo の設計と実装, SACSIS 2003 (先進的計算基盤システムシンポジウム), pp.113-120 (2003).

(平成 17 年 1 月 24 日受付)

(平成 17 年 4 月 30 日採録)



小野 功 (正会員)

平成 6 年東京工業大学工学部制御工学科卒業, 平成 9 年同大学大学院総合理工学研究科知能科学専攻博士後期課程修了。博士 (工学)。東京工業大学大学院総合理工学研究

科リサーチアソシエート, 徳島大学工学部助手, 同大学同学部講師, 同大学同学部助教授を経て, 平成 17 年より東京工業大学大学院総合理工学研究科助教授。主に進化計算, 人工知能, グリッドの研究に従事。IIZUKA'96 Student Paper Award, 日本光学会光設計グループ光奨励賞, 計測自動制御学会論文賞を受賞。計測自動制御学会, システム制御情報学会, 日本光学会各会員。



水口 尚亮

平成 15 年徳島大学工学部知能情報工学科卒業，平成 17 年同大学大学院工学研究科知能情報工学専攻修士課程修了，平成 17 年 4 月より（株）富士通に勤務，現在に至る．主に進化計算，グリッドの研究に従事．



中島 直敏

平成 14 年徳島大学工学部知能情報工学科卒業，平成 16 年同大学大学院工学研究科知能情報工学専攻修士課程修了，平成 16 年同大学院同研究科情報システム工学専攻入学，平成 17 年東京工業大学大学院総合理工学研究科知能システム科学専攻編入学，現在に至る．主に，進化計算，バイオインフォマティクスの研究に従事．



小野 典彦

昭和 54 年東京工業大学理学部情報科学科卒業，昭和 56 年同大学大学院総合理工学研究科システム科学専攻修士課程修了，昭和 56 年（株）日立製作所入社，昭和 57 年同社退職，昭和 61 年東京工業大学大学院総合理工学研究科システム科学専攻博士課程単位取得退学．工学博士．広島大学工学部助手，徳島大学工学部助教授を経て，平成 9 年より同大学同学部教授．主に，人工知能，マルチエージェントシステム，進化的学習，強化学習等の研究に従事．



中田 秀基（正会員）

平成 2 年東京大学工学部精密機械工学科卒業．平成 7 年同大学大学院工学系研究科情報工学専攻博士課程修了．博士（工学）．同年電子技術総合研究所研究官．平成 13 年独立行政法人産業技術総合研究所に改組．現在同所グリッド研究センター主任研究官．平成 13 年より東京工業大学客員助教授を兼務．グローバルコンピューティング，並列実行環境に関する研究に従事．



松岡 聡（正会員）

昭和 61 年東京大学理学部情報科学科卒業，平成元年同大学大学院博士課程中退．同大学情報科学科助手，情報工学専攻講師，東京工業大学情報理工学研究科数理・計算科学専攻助教授を経て，平成 13 年より東京工業大学学術国際情報センター教授．平成 14 年より国立情報学研究所客員教授併任．博士（理学）．高性能システム，並列処理，グリッド計算，クラスタ計算機等の研究に従事．平成 8 年度情報処理学会論文賞，平成 11 年情報処理学会坂井記念賞受賞．ACM OOPSLA'2002，IEEE CCGrid2003 を含む多くのプログラム・大会委員長を歴任．グリッド国際標準化団体 Global Grid Forum の Area Director．ACM，IEEE-CS 各会員．



関口 智嗣（正会員）

昭和 57 年東京大学理学部情報科学科卒業，昭和 59 年筑波大学大学院理工学研究科修了，同年工業技術院電子技術総合研究所入所．以来，データ駆動型スーパーコンピュータ SIGMA-1 の開発，ネットワーク数値ライブラリ Ninf，クラスタコンピューティング，グリッドコンピューティング等に関する研究に従事．平成 13 年独立行政法人産業技術総合研究所に改組，平成 14 年 1 月より同所グリッド研究センター長．市村賞，情報処理学会論文賞受賞．グリッド協議会会長．SIAM，IEEE，つくばサイエンスアカデミー各会員．



楯 真一

昭和 60 年東京大学薬学部卒業，平成元年東京大学大学院薬学系研究科博士課程単位取得後退学．東京都臨床医学総合研究所（研究員），東京都立大学理学部化学科（助手），北陸先端科学技術大学院大学新素材センター（助教授）を経て，現在，生物分子工学研究所機能制御研究部部長．博士（薬学）．NMR 分光法，構造生物学の研究に従事．日本分光学会論文賞．日本生化学会，日本生物物理学会，日本分光学会，日本化学会各会員．