Ninfによる遠隔計算資源アクセスシステムの構築とグローバルコンピューティングシステムの性能評価

佐藤三久[†1] 田中良夫[†1] 草野和寛[†1]
中田秀基[†2] 関口智嗣[†2]
長嶋雲兵[†4] 松岡聡[†3]

グローバルコンピューティングのためのミドルウエアである、ネットワーク情報ライブラリNinfを用いて、広域のネットワークで接続された計算機を利用するいくつかのシステムを試作している。Ninfにより、遠隔の計算資源をクライアント側のプログラムの一つの計算コンポーネントとして利用できるようになる。プロトタイプとして、数値流体解析コンポーネントnetCFDと分子軌道計算コンポーネントnetMOを開発した。流体計算では、途中結果の大量のデータの出力が必要となることがあるが、計算とデータの入出力をオーバーラップさせることによって、遠隔の計算資源でも効率的に利用できる。netCFDを利用したデモシステムとして、Java appletを用いて広域ネットワーク上のWebブラウザから利用可能なシステムを構築した。appletはサーバ上でNinfインタフェースでサービスされている流体解析解析プログラムを起動して計算を行い、call backのインタフェースにより計算途中でappletの可視化ルーチンを起動することで可視化させることができる。

Development and preliminary evaluation of remote computing resource access systems using Ninf

Mitsuhisa Sato ,[†1] Yoshio Tanaka ,[†1] Kazuhiro Kusano ,[†1]
Hidemoto Nakada ,[†2] Satoshi Sekiguchi ,[†2]
Umpei Nagashima [†4] and Satoshi Matsuoka[†3]

We are developing prototype systems to access remote computing resources by using a global computing middle ware, Ninf. Ninf allows the users to make use of the remote computing resources as computational components in his program. As our prototypes, we designed a Computational Fluid Dynamics (CFD) component, netCFD, and a Computational Chemistry component, netMO for molecular orbital computation. Though a large amount of data in each time step may be stored in CFD applications, the overhead of the I/O can be reduced by overlapping I/O and computation even in the remote CFD computation. As a demonstration of netCFD, we design the GUI using Java applet to make use of the netCFD component through Web browsers. The GUI applet invokes the CFD computation in remote Ninf server, and receives the results by the call back interface in Ninf to visualize the results in each time step.

1. はじめに

コンピュータネットワークの発展に伴い、ネットワークを利用した情報提供サービスが盛んに行われている。WWWなどの現在のネットワークサービスは、主にデータ自体の共有化であるが、これを推し推めることにより、計算に必要な資源、すなわちCPUなどの計算に必要な資源をも広域のネットワークにより仮想的に共有することが考えられる。最近、このような広域のネットワークの計算資源を使った計算システム、いわゆるグローバルコンピューティングが注目が注目されて始めている。

我々は、このような情報計算資源の共有のための枠組として、Ninf (Network based Information library for High Performance Computing)[4] を開発している。Ninfの目的は、LANあるいは広域のネットワー

†1 新情報処理開発機構
Real World Computing Partnership
†2 電子技術総合研究所
Electrotechnical Laboratory
†3 東京工業大学
Tokyo Institute of Technology
†4 物質工学工業技術研究所
National Institute of Materials and Chemical Research

ク上の数値計算ライブラリや科学技術計算に必要な数値情報データベースを通じて、おもに科学技術計算分野の情報ならびに計算資源を提供・共有する仕組みを提供することである。

広域のネットワークを利用して高性能計算を行うためのグローバルコンピューティングシステムはその利用形態・目的から、以下の 2 つに大別することができる：

- 複数の計算資源上で分散・並列計算を行うシステム。 globus[2] 上の MPI、 Nexus 等はこれに当たる。大規模なスーパーコンピュータ等を結合し、計算資源、計算規模の拡大を目的とする。
- 手元にある計算機でユーザインタフェースを提供し、遠隔の計算機を利用するシステム。このようなシステムは、 Datorr(Disktop Access to Remote Resource)[1] と呼ばれ始めており、手元にある PC や WS から、ネットワークで結合された大規模な計算機システムに" シームレス" に接続することを目的とする。

本稿では、後者のシステムに注目する。 Ninf システムは、グローバルコンピューティングのためのミドルウエアであり、もちろん前者にも適用可能であるが、 RPC(Remote Procedure Call) をベースに簡便なインタフェースを提供しており、手元の PC や WS から、簡単に遠隔の計算資源を利用できるようになる。 Ninf は、クライアントのプログラムにおいて、遠隔の計算で実行されるルーチンを一つのコンポーネントとし利用する手段を提供する。

我々は、 Ninf を用いて遠隔の計算資源を利用するアプリケーションとして、数値流体計算のためのコンポーネント netCFD と分子軌道計算のためのコンポーネント netMO を試作した。これを用いるデモシステムとして、 Java applet を用いて Web ブラウザから試用するユーザインタフェースを構築した。

科学技術計算システムが大規模かつ複雑化し、高性能計算機の形態が多様化する中で、そのソフトウエアはモジュール化、コンポーネント化する必要に迫られている。例えば、並列計算機は高性能化する有効な手段であるが、ユーザ側にとっては、並列化されたルーチンを利用するために、新たにそのルーチンを用いているプリ・ポスト処理等のプログラムの書き換えなどの作業が必要なる。 Ninf により、統一化されたインタフェースを持つ計算コンポーネントとすることにより、計算資源が逐次計算機であっても、並列計算機でもクライアントは同様なインタフェースで利用できるようになる。実際、今回試作した netCFD では逐次の数値流体計算プログラムの他に、クラスタ等並列環境で実行する並列版のプログラムも作成し、クライアント側からは、同様のインタフェースで呼び出すことができるようにした。このように、 Ninf を用いて利用する計算サーバを並列機などの高性能コンピュータにすることによって、高性能な計算資源を広域ネットワーク環境で手軽に利用できるようになる。

これまでにも、ネットワーク経由で計算資源にアクセスするシステムが構築されている。 PAPIA[6] は、 Web の CGI を用いて、並列計算機を利用してタンパク質の高速検索などの支援を行うシステムである。 Virtual Micro Scope(VRMS)[5] では、計算化学の大規模計算と GUI を PVM を用いて統合している。しかし、これらのシステムでは専用のインタフェースあるいは専用のプロトコルを用いており、クライアント側において任意のプログラムの一部として利用することが困難である。

また、分散処理ソフトウエアのコンポーネント化を行うミドルウエアとして、 CORBA や Java の RMI、 HORB などがある。 CORBA は、主にビジネスアプリケーションをターゲットとしたものであるが、最近では科学技術計算分野のソフトウエアにも応用されている例もある。 RMI は Java の分散オブジェクトの仕組みであるが、これと JINI を用いることにより、 Java 以外のソフトウエアにも応用可能である。

次章において、試作した数値流体計算コンポーネント netCFD、次に、分子軌道計算コンポーネント netMO について述べる。今回、 netCFD の簡単な性能評価を行ったので、 4 章において、報告する。 5 章において、今回の試作を通じて明らかになった課題、問題点について議論し、まとめる。

2. netCFD: Ninf による数値流体解析のための計算コンポーネント

2.1 概　　要

Ninf のアプリケーションの一つとして、数値流体解析プログラムの主要部分を Ninf のインタフェースを通じて利用できる計算コンポーネント netCFD を試作した。 netCFD に含まれるルーチンは、以下のものである：

- 数値流体解析の行うルーチン。引数で指定された時間ステップにおいて、ステップ毎のデータを引数にしてクライアント側を call back する機能を持つ。
- 利用するルーチンを制限するためのユーザ管理ルーチン☆
- デモ用のシミュレーションを行う空間を定義するメッシュデータをロードするためのルーチン

流体解析を行うプログラムは、我々が開発した有限要素法による並列 3 次元流体解析プログラム femFlow[7] である。その仕様を示す:

- 基本方程式として、連続の式、運動方程式 (ナビエストークスの式)、エネルギー保存式を解く。
- 離散化は有限要素法。 8 点のアイソパラメトリック要素。 3 次風上差分。時間積分法は、 SMAC 法。
- ソルバーは、 SCG 法。並列化に対応。

☆ 現在のバージョンは、ユーザ管理は Ninf 自体ではなくアプリケーションレベルで行っている。

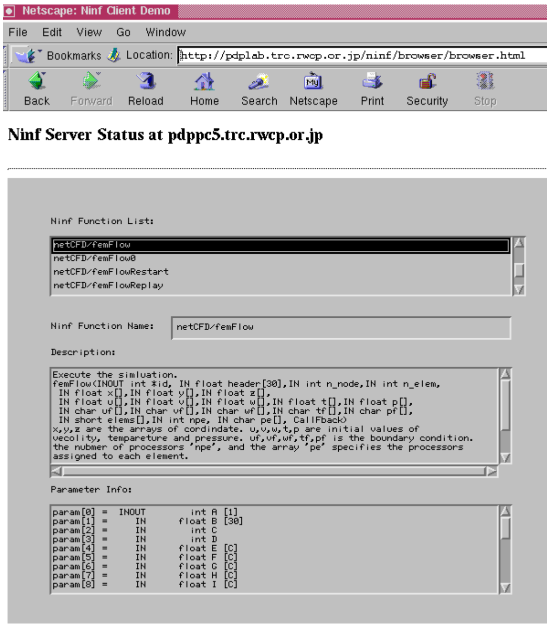

図 1　NinfBrowser

- C++ によるオブジェクト指向実装。

並列計算機においては、並列実行による高性能化が可能である。並列プログラムは、 MPI を用いて実装されており、ホストプログラムとは TCP/IP により簡単なプロトコルでデータのやりとりを行う。

各ルーチンのインタフェースは、 Ninf IDL で記述されており、 Ninf IDL コンパイラによりスタブルーチンを作成し引数に応じたプロトコルが自動的に生成される。例えば、計算の中心部分であるソルバーを呼び出す部分 femFlow は、 Ninf IDL により、以下のように定義されている：

```
Define femFlow(IN int id, IN float header[30],
 IN int nn,IN int ne,
 IN float x[nn],IN float y[nn], IN float z[nn],
 IN float u[nn],IN float v[nn], IN float w[nn],
 IN float t[nn],IN float p[nn],
 IN char uf[nn],IN char vf[nn],IN char wf[nn],
 IN char tf[nn],IN char pf[nn],
 IN short elems[ne*8],IN int npe, IN char pe[ne],
 flowDisplay(IN int step[1],IN double ctime[1],
    IN double delta[1],IN float data[nn*5])){
             ソルバーの呼び出し...
}
```

IN で指定されているパラメータは、 Ninf サーバを通じて、流体数値計算のメインプログラムに送られる。x,y,z はメッシュノードの座標、elem はメッシュ要素の指定、u,v,w,t,p 等は初期値である。並列実行のために、各メッシュ要素のプロセッサの割り当てをpe で指定している。最後に指定されているパラメータは、call back のインタフェースであり、指定されたステップで、クライアント側のルーチンを呼び出す。このインタフェースを用いて、各ステップのデータの取得、可視化を行うことができる。

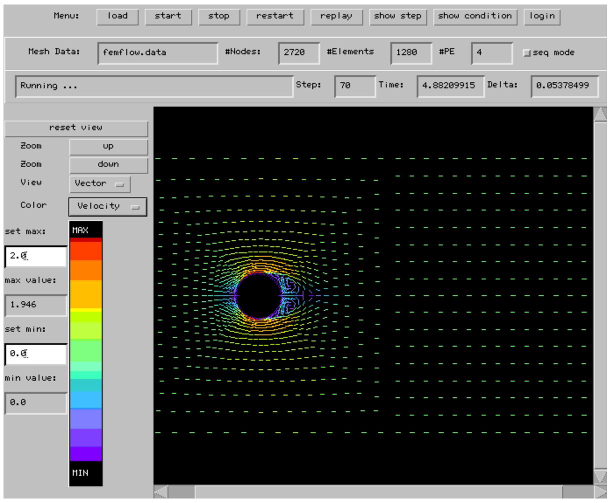

図 2　netCFD demo GUI (showflow)

指定されたサイトで利用可能な Ninf ライブラリは、Web 上の applet である NinfBrower によって、検索することができる。図 1に、そのインタフェースを示す。Ninf ライブラリは、 C、 Fortran, Java などから、各言語において提供されている Ninf_Call クライアントライブラリを用いて、呼び出すことができる。

2.2　Java Applet による GUI の設計

netCFD による計算コンポーネントの利用するデモアプリケーションとして、 Java applet による GUI を設計した。このアプレットにより、 Web ブラウザから、メッシュデータを選択し、数値流体計算を起動、実行結果を可視化することができる。 2 次元のベクトル図、色によるノード、要素の流速、圧力、温度の表示など、簡単な可視化機能がサポートされており、ユーザは実行中に確認して、シミュレーションを対話的に進めることができる。図 2にその GUI を示す。

もちろん、流体解析プログラムに特化した同様なシステムは構築可能であるが、 Ninf インタフェースを用いることにより、 Ninf クライアントライブラリさえあれば、自分のシステムの一部として、繁雑なネットワークプログラミングなしに利用できる。

我々はスーパーコンピューティングとネットワークの国際会議である SC'98 において PC を高速ネットワークで結合した RWC PC Cluster を netCFD の一つのコンポーネントとするデモンストレーションを行った。図 3にその構成を示す。これにより、スーパーコンピュータなどの高性能な計算機はもちろん、最近高性能計算のためのプラットフォームとして注目されている PC/WS クラスタもグローバルコンピューティングのための計算資源として簡単に利用できるようになる。

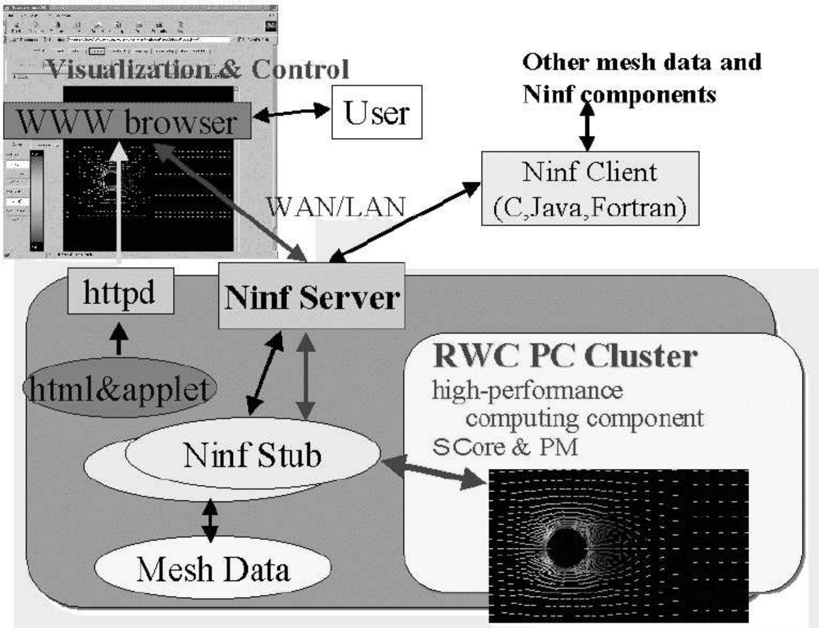

図 3　netCFD Demo (at SC'98)

3. netMO: 分子軌道計算のための計算コンポーネント

netMO は分子軌道計算プログラム Gamess を利用した分子軌道計算のための Ninf コンポーネントである。分子軌道計算のプログラムとしては、 Gamess の他にも有名な Gaussian や AMOSS があるが、この様なアプリケーション分野では、プログラムがソフトウエアとして完結しており、ルーチンとして、切り出すことが難しいことが多い。そこで、 Ninf を用いたインタフェースとして、入力ファイルの内容を文字列としておくり、結果ファイルも文字列として得ることにした。 Ninf サーバで実行されるルーチンは、入力の文字列をファイルに書き込み、必要な環境をセットアップし、 Gamess を起動する。実行終了後、得られた結果ファイルを文字列として返す。クライアント側では、結果ファイルを解読して、固有値、最終エネルギなどの値をプログラム中で利用する。

図 4に、デモシステムとして試作した Java applet による GUI を示す。このようなシステムは Web の CGI を用いても構築できるが、 Ninf インタフェースを用いることによって、 Java や C などのプログラムで、利用できるようになる。現在、入力データの生成、結果のグラフィック表示などのシステムを加えることを計画している。

4. netCFD の性能評価

グローバルコンピューティングシステムの性能評価の一例として、 netCFD について、性能測定を行った。測定環境として、新情報処理開発機構つくば研究センタの

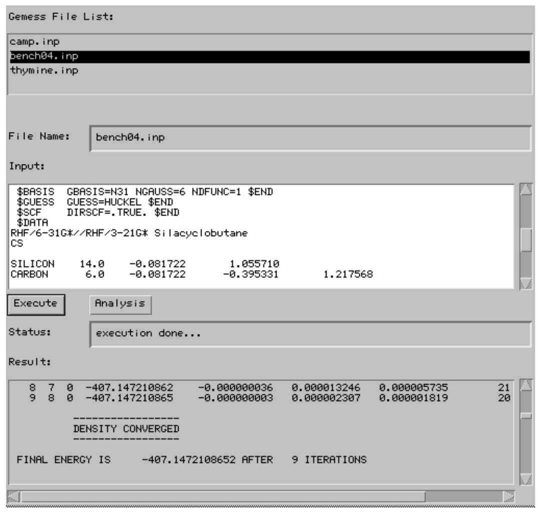

図 4　netMO GUI

SUN Ultra Enterprise 450 (4 CPU, 300MHz Ultra SPARC) に Ninf サーバを置き、電総研の SUN Ultra Enterprise (6 CPU, 336 Hz, Ultra SPARC) においてクライアントを実行した。新情報と電総研 (etl) は、IM net(1.5Mbps) 経由で接続されている。 Ninf サーバの基本性能を測定するために、以下のインタフェースを持つダミールーチンを用意した。

```
Define test(IN int in_size, IN int input[in_size],
       IN int out_size, OUT int output[out_size])
{ /* do nothing */  }
```

このルーチンは、in_size で指定されたサイズの配列を転送し、out_size で指定された配列を受け取る。out_size を 1 に固定し、in_size を変化さ

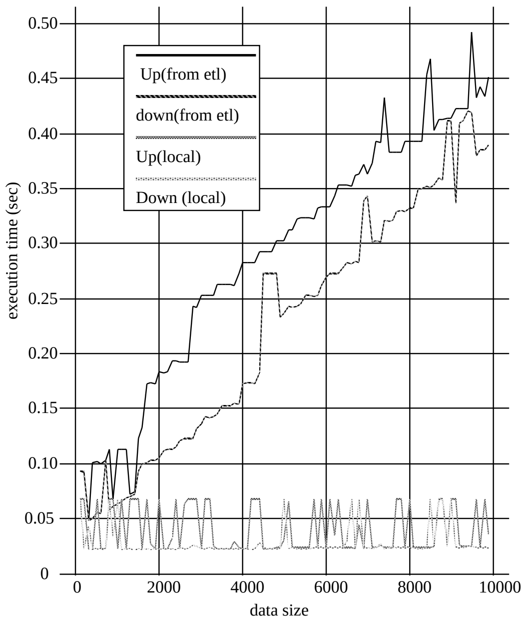

図 5　Ninf の基本性能

せた`Ninf_call`の実行時間 (Up)、`in_size`を 1 に固定し、`out_size`を変化させた Ninf_call の実行時間 (Down) を測定した。数回実行した内の最小時間を図 4に示す。 Ninf サーバと同じマシンで実行した場合は、 0.05 秒とほとんど変化が無いが、 etl からの場合は Up が Down を上回ることが観測された。スタブプログラムの起動等、オーバヘッドは約 0.05 sec、バンド幅は、 0.2Byte/sec である。

典型的な CFD のアプリケーションして、各時間ステップのメッシュ上のノードの流速、温度、圧力をファイルに格納するプログラムを考える。テストデータとして、 2 種類の円柱の周りのメッシュのデータ Small(2720 ノード、 1280 要素)、 Large(10560 ノード、 5120 要素) を用意した。

Ninf サーバを経由せずに直接実行した場合 (local), 同じマシン上で Ninf サーバ経由で実行した場合 (ninf(local))、 etl から ninf サーバ経由で呼び出した場合 (ninf(etl)) の実行時間を測定した。さらに、 Ninf サーバからはイーサネットで結合した Pentium Pro 200MHz 4 台のクラスタ上の並列 CFD プログラムを実行できるようにし、並列化プログラムによる実行時間も測定した。図 6は、それぞれ 10 ステップ、 100 ステップ、 200 ステップの実行時間である。 Ultra は、サーバの実行されているマシンでの実行時間、 Pentium*4 はクラスタで並列実行した時の実行時間である。 Ninf サーバ経由で実行する場合は、各時間ステップのデータは各ステップの終了時点で、ファイルに各ステップのデータを書き出す call back ルーチンを呼び出すことで、クライアントに送られる。サーバでの実行は、 call

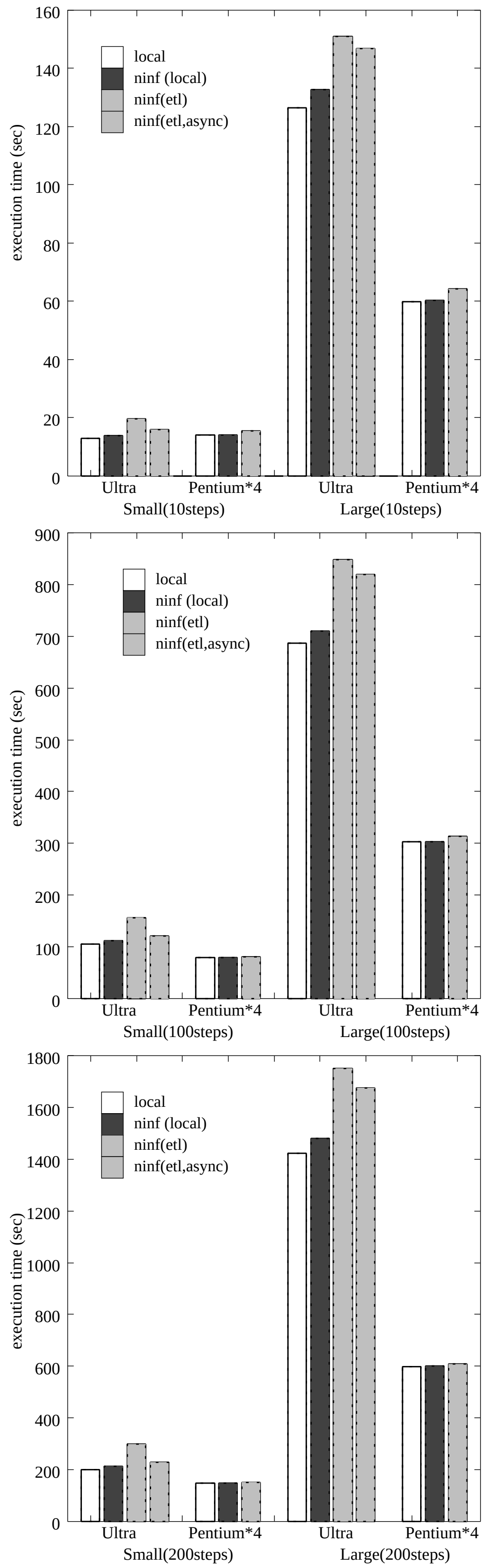

図 6　netCFD の性能

back の呼び出しが終了するまで待つため、単にデータを送出するこの場合は不要な待ち時間が生じる可能性がある。そのため、 call back 終了を待たないモードでの実行時間も測定した (ninf(etl,async))。並列 CFD エンジンを用いる場合には、サーバのルーチンが call back の終了を待つ間も並列プログラムは実行を中断しないので、この必要はない。なお、各ステップで送られるデータ量は、 Small 106 KB、 Large 412 KB である。

全体として、 Ninf を経由して遠隔クライアントで実行しても、そのオーバヘッドは、 10% から 20% である。このアプリケーションのように各ステップにおいて比較的大量のデータを転送する場合でも、通信が隠蔽されていれば、遠隔実行でも十分効率的である。

なお、初期境界条件からの実行では最初の数ステップにおいて実行時間がかかる。また、並列 CFD エンジンを用いる場合には MPI プログラムの起動に時間がかかるため、実行ステップが長くなればなるほど、 Ninf による起動等のオーバーヘッドが相対的に小さくなる。

5. 課題・問題点

実用的に運用するために、まず第一に解決しなくてはならない問題点は、ユーザの認証である。今回のシステムはデモのためのシステムであるため、 netCFD のルーチンに簡単な認証のルーチンを組み込むだけにとどめている。これは Ninf システムが提供することが望ましい。 Ninf の認証については、既に提案されており[8]、それを組み込む必要がある。セキュリティを高めるためのデータの暗号化については、そのオーバーヘッド等の評価を進める必要がある。

計算資源のスケジューリングについては広域の計算資源を選択するスケジューリングとサーバー内でのスケジューリングの 2 つのレベルがある。 Ninf では、広域の計算資源の中で同様なコンポーネントを提供しているサーバ間は、メタサーバ[3] によって、スケジューリング機能を提供している。今回試作したシステムでは、メタサーバーによるスケジューリング機能などを使っていないが、メタサーバを用いることによって、同じ計算サービスが複数のサイトで行われている場合、適宜選択してくれるようになる。サーバ内においては、 Ninf 経由のタスク間のみならず、ローカルなタスクとのスケジューリングを考慮しなくてはならない。実用的な運用を考えると、 TSS で実行されない並列機の場合や長時間実行されるタスクなどは特別な配慮が必要となる。

また、デモシステムでは Java applet を用いたが、アプレットではローカルなファイルにアクセスできない、ダウンロードホスト以外と通信できないなど制限がある。デモにとっては、手元にプログラムがある必要がないなどメリットが大きいが、実用的上を考えた場合にはダウンロードしてもらい、 Java アプリケーションとして実行してもらうことになるであろう。

6. おわりに

試作したシステムは、デモシステムにとどまっており、実用に供するにはインタフェースの設計、セキュリティ、ジョブのスケジューリングなど、実際のアプリケーションユーザの利用を通じて改善していく必要がある。高性能必要とするユーザにはまだ十分でないものの、教育向けには活用できると考えている。 netCFD については、メッシュなどをユーザーが設定できるシステムを構築する計画である。 netMO については、分子軌道解析ための GUI を考えている。

現在、試用環境として、
`http://pdplab.trc.rwcp.or.jp/`
において、 netCFD のデモ GUI を公開をしているので、興味のある方はアクセスを試みられたい。

謝辞：本研究を遂行するにあたり Ninf プロジェクトメンバー諸氏に感謝いたします。本研究の一部は、電子技術総合研究所において、科技庁原子力特別研究「複雑現象解明における超高速計算機利用技術の研究」で行われたものである。

参 考 文 献

1) Notes of the 1st Intl. Workshop on 'Desktop Access to Remote Resources'. http://www-fp.mcs.anl.gov/ gregor/datorr.
2) I. Foster and C. Kesselman. Globus: A metacomputing infrastructure toolkit. In *Proc. of Workshop on Environments and Tools*. SIAM, 1996. http://www.globus.org/.
3) H. Nakada, H. Takagi, S. Matsuoka, U. Nagashima, M. Sato, and S. Sekiguchi. Utilizing the Metaserver Architecture in the Ninf Global Computing System. In *Proc of HPCN'98 (LNCS 1401)*, 1998.
4) M. Sato, H. Nakada S. Sekiguchi, , S. Matsuoka, U. Nagashima, and H. Takagi. Ninf: A Network based Information Library for Global World-Wide Computing Infrastructure. *Proc. of HPCN'97 (LNCS 1225)*, pages 491–502, 1997.
5) 関口, 高田, 松原, 逢坂, 平原. バーチャルマイクロスコープのワークステーションクラスタによる高速実装. In 情報処理学会研究会報告 *HPC98-71-1*, 1998.
6) 鬼塚, 野口, 斎藤, 秋山. タンパク質立体構造研究支援のための並列統合解析システムの構築. In 情報処理学会研究会報告 *HPC97-68-8*, 1997.
7) 西川, 佐藤, 田中. 並列非圧縮性流体解析プログラムのオブジェクト指向言語による実装. In 日本流体力学会年会 *'98* 講演論文集, 1998.
8) 中田, 松岡, 佐藤, 関口. 広域計算システム Ninf におけるユーザ認証. In 情報処理学会研究会報告 *HPC98-72-14*, 1998.