情報処理学会 研究報告,
2003-HPC-93-(29), pp. 167-172,
2003年 3月 12日

気象予報グリッドポータルの開発

首藤 一幸[†]　武宮 博[†]　平野 基孝[††]　田中 良夫[†]　関口 智嗣[†]

気象予報を行なう既存の逐次プログラムをグリッド上で動作するように Ninf-G を用いて修整し、産総研で開発しているポータル構築キット GridLib に組み込み、気象予報グリッドポータルを開発した。既存プログラムの Ninf-G 対応、GridLib への組み込みともに容易であり、特に GlidLib 対応においてはプログラムの改変や新規開発は一切不要である。両ミドルウェアを利用することで、少ない労力でグリッドポータルを構築できた。本稿ではこのグリッド化、ポータル化の経験を報告する。

A Grid Portal for Numerical Weather Prediction

Kazuyuki Shudo[†]　Hiroshi Takemiya[†]　Motonori Hirano[††]
Yoshio Tanaka[†]　Satoshi Sekiguchi[†]

This paper introduces our experience on development of a Grid Portal, which provides a weather forecast service via web browsers to its users. The development consists of two parts, as “Gridification” of a forecasting program with Ninf-G and its “Portalization” with GridLib. We could accomplish those work with a little deal of trouble. Especially, any modification to the program is not necessary to portalize it.

1　はじめに

広域に分散した管理主体の異なる計算機群を束ねて集合的に活用することは、グリッドというコンセプトの主要な一面である。すでに、PKI を活用してのセキュアな遠隔プログラム起動や通信を可能とする下位グリッドミドルウェア Globus Toolkit [1] や、その上で容易なプログラミングを支援する MPICH-G2 [4]、Ninf-G [6, 5] といった上位ミドルウェアが提案されてきており、このコンセプトは現実のものとなりつつある。

これらの取り組みは、様々なリソースを容易にセキュアに組み合わせて活用するという便宜を、アプリケーション開発者に対して提供するものである。一方、これらの便宜をアプリケーション利用者にまで直接提供しようという取り組みのひとつとして、グリッドポータル（portal）が注目されている。グリッドポータルとは一般に、ウェブブラウザを用いて容易にグリッドアプリケーションを利用できるようにするシステムである。グリッドポータルには共通して必要となる機能が多くあるため、これら共通の機能を提供するポータル構築キットが提案されてきた [10, 2, 12]。

既存のポータル構築キットの多く [2, 10] は、アプリケーションをポータルから利用可能とするために、Perl や JavaServer Pages などを用いたプログラミング作業を必要とする。これに対して、産総研では、既存のアプリケーションを一切の変更なしにポータルへと仕立てることのできるグリッドポータル開発キット GridLib を開発している。これは、ASP（アプリケーションサービスプロバイダ）としての仮想スーパーコンピュータセンターを構築することを目的としたミドルウェアである。

我々は、逐次プログラムのグリッド化およびポータル化を行うことで、それぞれを支援するミドルウェア Ninf-G と GridLib の利便性、有用性を評価した。具体的には、既存の気象予報プログラムを、Ninf-G を用いた非同期遠隔呼び出しを行うように修整し、GridLib に組み込むことでポータルに仕立てた。

本稿ではこの経験を報告する。続く 2 章では本ポータルの概要を述べ、3 章にて GridLib について説明する。4 章では気象予報モデルとグリッド化に適した予報精度の向上手法、また、予報プログラムの Ninf-G 対応について述べる。5 章では、開発者から見た GridLib の有効性を論じ、実行例を示すことで利用者から見た本ポータルの利便性を示す。

2　概要

本アプリケーションポータルは、図 1 に示すいくつかのソフトウェアで構成されている。

GridLib は、グリッドポータルを容易に開発するためのミドルウェアである。

[†]産業技術総合研究所　National Institute of Advanced Industrial Science and Technology
[††]株式会社 SRA
Software Research Associates, Inc.

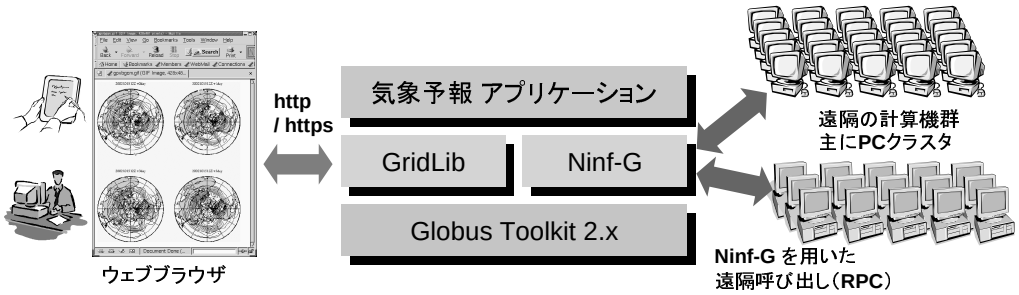

図 1: 気象予報ポータルのアーキテクチャ

気象予報アプリケーションは、Ninf-G を用いた遠隔呼び出しで分散処理を行うようにグリッド化した。Ninf-G は Globus Toolkit 2.x [1] のセキュリティ機能と通信機能を用いて遠隔呼び出しを行うグリッドミドルウェアであり、Ninf プロジェクトのウェブページ [7] から入手可能である。

3 グリッドポータル開発キット GridLib

GridLib は、我々が開発を進めている、ASP としての仮想スーパーコンピュータセンターを構築するためのポータル開発キットである。既存のアプリケーションを、何ら変更を加えることなく、わずかな手間で、ウェブブラウザから利用できるようにポータル化できる。Gaussian Portal [11] など、気象予報以外の他のアプリケーションのポータル化にも利用されている。GridLib 自身は JavaServer Pages（JSP)、Servlet、Java 言語で記述されたいわゆるウェブアプリケーションである。

以下に挙げる、ポータルに共通して要求される機能の一切は GridLib が提供する。

- サインオンの受け付け
  パスワードと、ウェブブラウザが提示する証明書に基づいて、利用者を認証する。
- ジョブの実行管理
  利用者の要求を受け、Globus Toolkit のジョブ管理機能 GRAM を利用し、ジョブの投入、取消といった制御を行う。また、実行中、完了などの状態を管理し、利用者に提示する。
- ジョブの入出力管理
  パラメータの入力やファイルの転送を受け付け、アプリケーションの起動時に引数として渡す。また、ジョブの出力（標準出力やファイル）を利用者に返す。
- アカウンティング
  プロセッサとディスクの使用量に基づく利用記録を採る。コンピュータセンターには必要な、実用には欠かせない機能である。

アプリケーションのポータル化は、アプリケーション記述ファイルを作成し、GridLib に登録することで完了する。この記述ファイルは、Ninf-Portal [12] の Grid アプリケーション IDL に相当する。例えば、GridLib に図 2 のアプリケーション記述を与えると、利用者のウェブブラウザには図 3 のパラメータ入力画面が提示される。

4 予報モデルとそのグリッド化

ポータル化した気象予報アプリケーションが用いている予報モデルを説明する。また、このモデルに対して適用したアンサンブル予報というグリッド向きの予報精度改善手法について述べる。

4.1 順圧 S–Model

本ポータルは、順圧 S–Model [8] という中・長期予報のための大気大循環モデルを用いて気象予報を行う。計算による気象予報では、予測モデルのカオス的な振る舞いが、精度良い中・長期予報を困難にしている。カオス的振る舞いの種となる誤差は、大気の観測や計算時の丸めにおいて不可避である。順圧 S-Model は、予測の対象を、大気の鉛直（重力）方向に平均化した状態量とすることで、カオス性を抑制し、予報限界を延ばしている。その結果、中・長期の気象に大きな影響を与えるブロッキング高気圧を精度良く再現でき、1 週間程度の中期予報に成功している。

4.2 アンサンブル予報

本ポータルの気象予報プログラムは、さらなる予報精度の向上を目的として、アンサンブル予報を行う（図 4)。アンサンブル予報では、大気シミュレーションの初期状態や各時刻の状態に対して、乱数から生成した擾乱を加える。それぞれ異なる乱数系列を用いて複数のシミュレーションを行い、得られた複数の結果の平均をとり、最終結果とするのである。これによって、カオス的振る舞いによって生じる予報のずれ抑制が期待できる。気象庁の場合、擾乱を加えた初期状態を 20 か

```
<?xml version="1.0" encoding=... ?>
<application
  ...
  <appname>ls</appname>
  <!-- -------- Command Line Spec. --- -->
  <argspec>/bin/ls %option% %width%</argspec>
  <!-- -------- Argument List -------- -->
  <arglist>
    <!-- ------ Required Args -------- -->
    <args use="required">
      <title>option</title>
      <radio name="option">
        <option value="-a">
          do not hide entries ...</option>
        <option value="-l">
          use a long listing format</option>
      </radio>
    </args>
    <!-- ------ Optional Args -------- -->
    <args use="option" option="-w">
      <title>screen width</title>
      <text name="width" size="14"
            maxlength="8" value="40">
        <constraint>
          <type value="integer" />
          <minExclusive value="0" />
          <maxInclusive value="200" />
        </constraint>
      </text>
      ...
    </args>
  </arglist>
</application>
```

図 2: GridLib のアプリケーション記述例

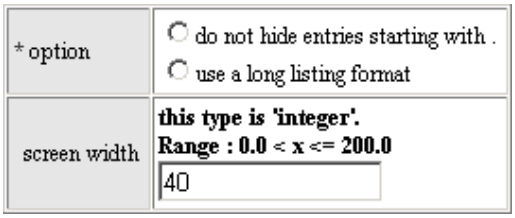

図 3: GridLib のパラメータ入力画面例

ら 50 通り作成してアンサンブル予報をしており、これによって 1 日程度、予報限界が延びたと言われている。

順圧 S-Model にアンサンブル予報を適用するにあたっては、初期状態には擾乱を加えず、大気シミュレーションの各時間ステップ毎に、外部 forcing に対して擾乱を加えて予測を行うようにした。

アンサンブル予報では、ひとつの結果を得るために数十のシミュレーションを行う必要がある。また、アンサンブル予報の効果を調べる研究段階では、一度に数百のシミュレーションを実行したい。このため、シミュレーション数に比例した多量の計算が一度に必要となるが、それぞれのシミュレーションは完全に独立して行える典型的なパラメータサーベイである。したがって、計算機間通信の遅延への耐性が高い。グリッドに向いた手法である。

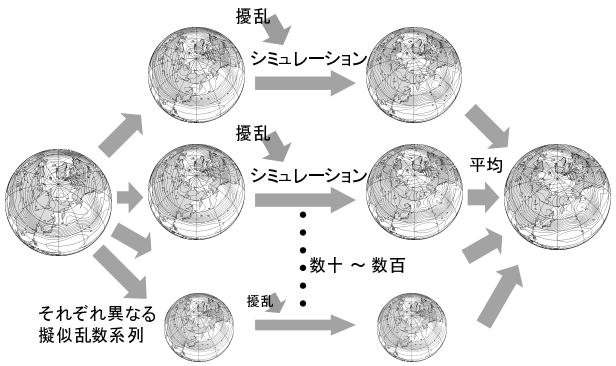

図 4: アンサンブル予報

```
Module S-model;
Define servmain(IN int N1, IN int IBUF[N1],
    IN int N2, IN double DBUF[N2],
    IN int N3, IN double WW[N3],
    IN int N4, OUT double TARRAY[N4],
    IN int N5, OUT double WTOT[N5])
Required "S-model_serv.o"
Calls "Fortran" servmain(N1, IBUF,
  N2, DBUF, N3, WW, N4, TARRAY, N5, WTOT);
FortranFormat "%s_";
```

図 5: 気象予報プログラムの遠隔呼び出しインタフェース記述

4.3 予報プログラムの Ninf-G 対応

この気象予報 FORTRAN プログラムを、Ninf-G を用いた遠隔呼び出しで分散処理を行うようにグリッド化した。Ninf-G 対応の手順は次の通りである。

1. 遠隔呼び出しの対象とする関数の決定
   独立して行うことのできる個々のシミュレーション（4.2 節）を分散処理の単位としてひとつの関数にまとめ、非同期遠隔呼び出しの対象とした。
2. caller・callee 間のデータ依存の除去
   Ninf-G では、遠隔呼び出しの caller、callee 間で共有できるデータは引数と返り値に限られる。そのため、非明示的に共有されるそれ以外のデータ、例えば C 言語の大域変数や FORTRAN の COMMON 変数は引数などを通して共有するようにプログラムを修整する必要がある。気象予報プログラムの Ninf-G 対応作業では、COMMON 変数を引数に変更した。
3. 関数呼び出しの遠隔呼び出しへの書き換え
   対象とする関数呼び出しを、Ninf-G の非同期呼び出し関数 `grpc_call_async` や呼び出し完了待ち関数 `grpc_wait_any` に置き換えた。

4. スケジューリングルーチンの作成
   現実装では、呼び出し先計算機を決定するメタサーバを利用していないので、気象予報の caller 側プログラムにセルフスケジューリングを実装した。
5. 呼び出しインタフェース定義ファイルの作成と MDS への登録
   遠隔呼び出しの対象とした関数について図 5 に示すインタフェース定義を記述して、Globut Toolkit の計算資源情報管理システム MDS に対して登録した。このインタフェース定義を Ninf-G の `ns_gen` コマンドに与えることで、遠隔呼び出しの callee 側スタブプログラム `_stub_XXX_.c` と MDS に登録する LDAP [3] データが生成される。

非同期呼び出しで行える並列処理は fork-join 型であるため、遠隔呼び出しがあらゆる並列処理に適するわけではない。しかし、fork-join 型の処理であれば、上記のように軽微な作業で Ninf-G を用いてグリッド化が可能である。現に、上記グリッド化作業においては、callee 側の処理は変更が不要であった。修整の分量も、caller 側の遠隔呼び出しへの書き換えは 10 行程度、スケジューラの実装は 50 行程度の追加で済んだ。

Globus Toolkit が提供する機能は計算資源情報管理機能（MDS）やセキュアな遠隔プロセス起動機能（GRAM）、バイト列の送受信機能（Globus IO）といったごく基本的なものでしかない。Globus Toolkit を直接用いてアプリケーションを開発することは多大な労力を要するだけでなく、バグが混入しやすく、また、デバッグも容易ではない。上位ミドルウェアである Ninf-G は、プログラマに対して遠隔呼び出しプログラミングモデルという枠を与えることで、アプリケーションの開発、保守、また、グリッド化を容易にしている。

5 評価と考察

ポータル開発者から見た GridLib の有用性と、利用者から見た気象予報ポータルの利便性を評価する。

5.1 GridLib を用いたグリッド化

ポータル化とは、具体的には、XML 形式のアプリケーション記述を作成して GridLib に登録するという作業である（3 章）。ポータル化のためにアプリケーション自体に手を加える必要は全くなかった。アプリケーション記述の作成も、図 2 のような例を参考にしてごく容易に行えた。

5.2 利用者から見た実行例

気象予報ポータル利用の様子を図 6 から図 8 に示す。アプリケーション利用者が本ポータルを利用する際の

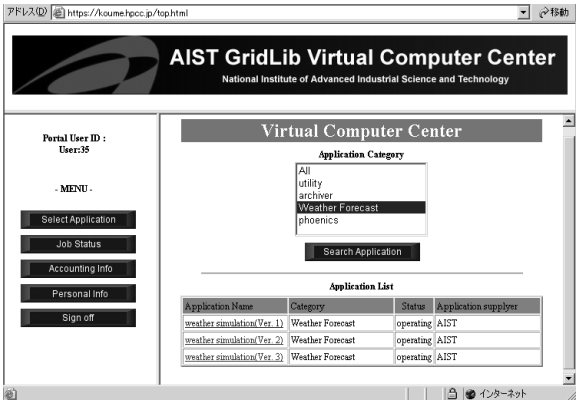

図 6: GridLib のアプリケーション選択画面

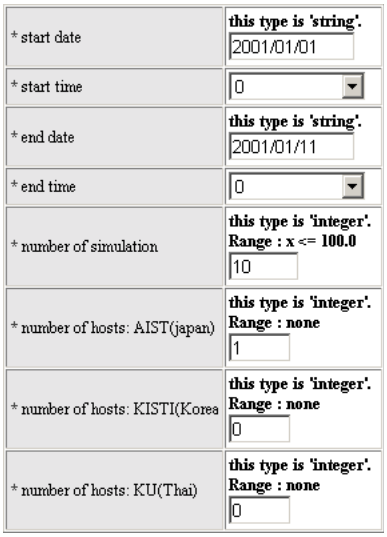

図 7: 気象予報のパラメータ入力画面

手順は、次の通りである。

1. ウェブブラウザに、ポータル管理者が発行した PKCS#12 などの形式の利用者の証明書を登録する。
2. ポータルのウェブページを閲覧し、パスワードを入力してサインオンする。
3. 選択画面（図 6）にて、アプリケーションを選択する。
4. そのアプリケーションに与えるパラメータの入力画面が表示されるので（図 7）、入力する。
5. [Job Status] ボタンを押すことでジョブ一覧を閲覧できる。投入したジョブが実行終了していれば、結果取得ボタンが現れている。
6. 結果取得ボタンを押すことで、ジョブが生成したファイルや、ジョブが標準出力（stdout）に出力した内容が記録されたファイルを取得できる。取得したアーカイブに可視化の結果（図 8）が含まれている。

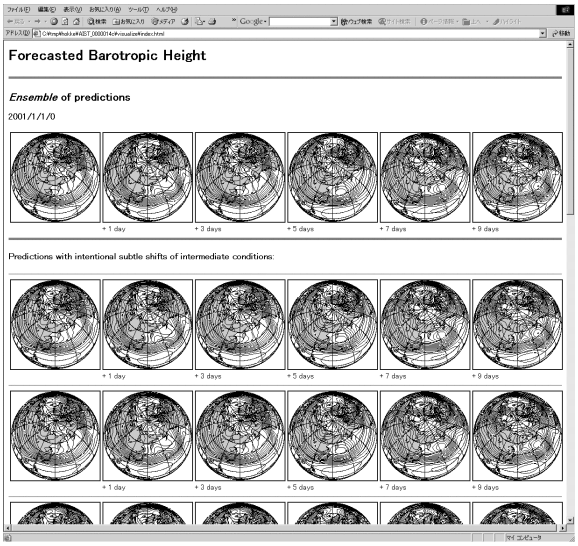

図 8: 気象予報の結果

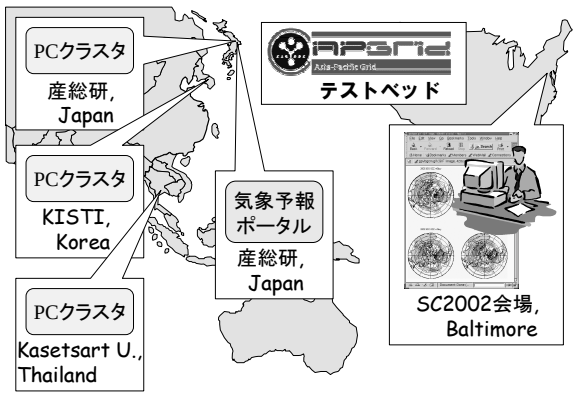

図 9: ApGrid テストベッドを用いた実験

ユーザインタフェースの部品はウェブブラウザで一般的なボタンやフォームであり、サインオンといった概念もウェブアプリケーションにおいて常識的なものである。そのため、ジョブなどの概念や、アプリケーションについての知識さえあれば、直観的に利用できる。

ジョブを投入するとジョブ ID が発行され、その時点でウェブブラウザは終了させることができる。後日、別の場所からジョブの実行結果を引き出すことも可能である。

5.3 広域テストベッドでの実験

本気象予報システムは現在、ApGrid [9] のグリッドテストベッド上で動作している。本ポータルから利用可能であるのは、テストベッドの計算機群のうち、産総研の Linux クラスタ 1 台、韓国 KISTI の Linux クラスタ 3 台、タイ Kasetsart 大（以下 KU）の Linux クラスタ 1 台、合計 6 台 158 プロセッサのクラスタ群である（図 9）。

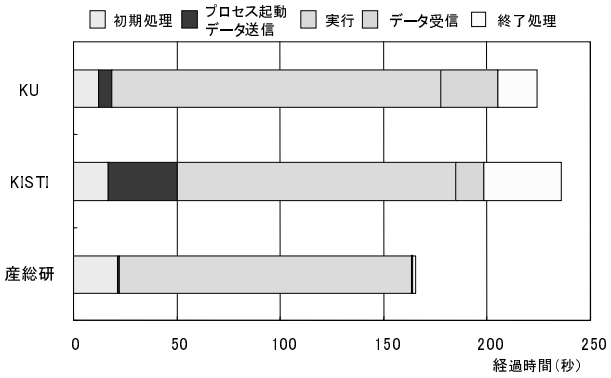

図 10: 遠隔呼び出しに要した時間

KU、KISTI、産総研それぞれの計算機 1 台に対して Ninf-G でシミュレーションを呼び出した場合の実行時間を図 10 に示す。100 日間の気象予報を行っており、シミュレーションの計算自体に要する時間はいずれの計算機でも 150 秒程度であった。しかし、Ninf-G での遠隔呼び出しには様々なオーバーヘッドが伴い、遠隔呼び出しの準備を始めてから結果が得られるまでの時間は、LAN において 15%増し、KISTI、KU への呼び出しでは 30~40%増しとなった。これは次の理由による。

- 初期処理のコスト：
  MDS へのアクセスに 10~20 秒を要する。
- GRAM を用いたジョブ起動のコスト：
  KISTI ではジョブ管理システム PBS を用いてバッチ運用を行っているため、GRAM を用いたジョブ起動要求から実際の起動までに 20 秒程度の時間を要する。
- データ転送のコスト：
  KISTI、KU への呼び出しでは、データの送受信にそれぞれ 10 秒程度を要する。
- 終了処理のコスト：
  Globus Toolkit のジョブ管理機能 GRAM が callee の終了を検知するのに 10~30 秒を要する。

これらのオーバーヘッドを削減するために、次の手法を検討している。

- MDS を利用せずに済むように、遠隔手続き情報を caller 側にファイルとして用意しておく。
- GRAM が起動したプロセスの終了を検知するために行うポーリングの間隔を短縮する。既定の間隔は 30 秒となっている。

6 まとめ

既存の逐次FORTRANプログラムのグリッドポータル化を通して、Ninf-Gを用いたグリッド化とGridLibを用いたポータル化の容易さを検証した。

まず、気象予報を行う逐次プログラムを、予報精度手法であるアンサンブル予報に着目して、Ninf-Gによる非同期遠隔呼び出しによってグリッド化した。グリッド化の作業は軽微なものであり、関数呼び出しの遠隔呼び出しへの修整とスケジューラの実装を合わせても、1日で完了した。

続いて、グリッド化したプログラムをポータル開発キットGridLibに登録し、アプリケーションポータルに仕立てた。GridLibは既存アプリケーションのポータル化支援に重点を置いており、ポータル化の作業ではアプリケーションに変更を加える必要は一切ない。アプリケーション記述ファイルを例にならって作成してGridLibに登録するだけで済む。ポータルのユーザインタフェースは一般的なウェブアプリケーションのそれであり、直観的な操作が可能である。また、ジョブの投入から結果の取得までサインオンを続けている必要はないため、ウェブブラウザさえあればどこからでもジョブ制御や結果の取得が行える。

ApGridテストベッドから3ヶ国の計算機群を利用し、実稼働の試験および性能評価を行った。グリッド化によるオーバーヘッドはときに40%にも及んだ。改善策を検討している。

今後は、気象予報モデルの研究を含めた実運用や、より大規模な実環境での実験を通じて、GridLib、Ninf-G、また、グリッドアプリケーションやポータル一般の課題を洗い出していく。

謝辞

気象予報プログラムを提供して頂くとともに、貴重な助言を頂きました筑波大学 田中博助教授に感謝致します。また、ApGridのテストベッドを構成する計算機を利用させて頂いたKISTI、Kasetsart大学の皆様に感謝致します。

参考文献

[1] Ian Foster and Carl Kesselman. Globus: a metacomputing infrastructure toolkit. *Int'l Journal on Supercomputer Applications*, Vol. 11, No. 2, pp. 115–128, 1997.

[2] DOE Science Grid. Grid portal development kit (GPDK). http://doesciencegrid.org/projects/GPDK/.

[3] Timothy A. Howes and Mark C. Smith. A scalable, deployable directory service framework for the internet. Technical report, Center for Information Technology Integration, University of Michigan, July 1995.

[4] Nicholas T. Karonis, Brian R. Toonen, and Ian Foster. MPICH-G2: A Grid-enabled implementation of the message passing interface. *Journal of Parallel and Distributed Computing*, to appear 2003.

[5] Hidemoto Nakada, Mitsuhisa Sato, and Satoshi Sekiguchi. Design and Implementations of Ninf: towards a Global Computing Infrastructure. In *Future Generation Computing Systems, Metacomputing Issue*, Vol. 15, pp. 649–658, 1999.

[6] Hidemoto Nakada, Yoshio Tanaka, Satoshi Matsuoka, and Satoshi Sekiguchi. *Grid Computing: Making the Global Infrastructure a Reality*, chapter Ninf-G: a GridRPC system on the Globus toolkit. John Wiley & Sons Ltd, 2003.

[7] Ninf Project. Ninf project home page. http://ninf.apgrid.org/.

[8] H. L. Tanaka and Daisuke Nohara. A studiy of deterministic predictability for the barotropic component of the atmosphere. In *The Science Reports of the Institute of Geoscience, University of Tsukuba*, Vol. 22A, pp. 1–21. March 2001.

[9] Yoshio Tanaka. Apgrid, 2001. http://www.apgrid.org.

[10] Mary Thomas, Stephen Mock, and Jay Boisseau. Development of web toolkits for computational science portals: The NPACI HotPage. In *Proc. of High Performance Distributed Computing (HPDC 9)*, pp. 308–309, August 2000.

[11] 西川武志, 長嶋雲兵, 関口智嗣. Quantum Chemistry Grid/Gaussian Portal Phase2. 情報処理学会研究報告, 2002-HPC-92(-8), pp. 43–48, October 2002.

[12] 中田秀基, 齊藤真幸, 鈴村豊太郎, 田中良夫, 松岡聡, 関口智嗣. Gridポータル構築ツールキット Ninf-Portal. JSPP2002 論文集, pp. 209–216, May 2002.