

# アプリケーションのインストール、データの配布、更新をサポートする グリッドポータル構築ツールキット (PCT4G) の開発

白 砂 哲<sup>†</sup> 鈴村 豊太郎<sup>†</sup>  
中 田 秀 基<sup>††,†</sup> 松 岡 聡<sup>†,†††</sup>

グリッド技術が実用化するにつれ、数多くのグリッドポータルと呼ばれるユーザ親和なインタフェースを備えたシステムがさまざまな分野で構築され利用されている。また、グリッドポータル構築者の負担を軽減するため、グリッドポータルを構築を支援するためのいくつかのツールキットも開発されている。しかし、それらのツールキットを利用しても、すべてのノードに利用するアプリケーションをインストールする必要があるなど、グリッド構築者の負担は大きい。更に、バイオインフォマティクス分野に代表されるアプリケーションでは、アプリケーションが利用するデータを最新の状態に保つ必要がある。これらの作業を自動化するため、我々はグリッドポータル構築ツールキット PCT4G を開発した。PCT4G は、アプリケーションのインストール、データの管理、Web インタフェース作成をサポートし、グリッドポータル構築者の負担を軽減する。また、PCT4G は、一般ユーザが Web インタフェースを用いて独自のグリッドポータルを構築するための機能も提供する。

## Development of A Grid Portal Construction Toolkit (PCT4G) Supporting Application Installation and Data Distribution/Update

SATOSHI SHIRASUNA,<sup>†</sup> TOYOTARO SUZUMURA,<sup>†</sup>  
HIDEMOTO NAKADA<sup>††,†</sup> and SATOSHI MATSUOKA<sup>†,†††</sup>

As Grid technologies become more practical, a number of Grid Portals have been constructed and used in various fields to offer user friendly interfaces for Grid resources. Along with that, several toolkits to generate Grid portals have been developed in order to reduce the burden of portal developers. However, even with the aid of those toolkits, portal developers still have to install target applications on each node on the Grid. In addition to that, it is necessary to keep application data up to date for some applications, especially applications in bioinformatics field. In order to automate these tasks, we are implementing a toolkit, PCT4G, which automates application installation, data management, and interface generation. Also, users can construct Grid Portals for their own applications on the fly through Web interfaces of PCT4G.

### 1. はじめに

グリッド技術の成熟にともない、グリッドの複雑さを隠蔽し豊富な計算力を簡便に利用するためのシステム、グリッドポータルが注目されている。グリッドポータルとは、通常インタフェースに Web を用い、一般的な Web ブラウザからグリッド資源の利用を可能にするシステムである。ユーザは、Web ブラウザを用

いてグリッドポータルにログインし、利用するアプリケーションを選び、引数やデータなどを入力する。グリッドポータルはさまざまな分野で実用されており、また、グリッドポータルを構築を支援するツールキットもいくつか開発されている。

しかし、既存のツールキットは、グリッドの複雑さを一般ユーザから隠蔽しているものの、グリッドポータル構築者への支援は十分ではない。例えば、グリッド上の無数の計算機へのアプリケーションのインストール、設定はポータル構築者が個別に行う必要がある。さらに、BLAST に代表されるバイオインフォマティクス分野のアプリケーションの多くは、最新に保たれたデータを必要とすることが多い。そのため、アプリケーションが利用するデータの配布や更新のコストもポータル構築者の大きな負担となっている。

<sup>†</sup> 東京工業大学

Tokyo Institute of Technology

<sup>††</sup> 産業技術総合研究所

National Institute of Advanced Industrial Science and Technology

<sup>†††</sup> 国立情報学研究所

National Institute of Informatics

これらのグリッドポータルを構築、保守する際のコストを削減するため、我々はポータル構築ツールキット”Portal Construction Toolkit for the Grid (PCT4G)”を開発した。PCT4Gは、アプリケーションのグリッド資源上へのインストール、アプリケーションが利用するデータの配布、更新、インタフェースの自動生成をサポートする。また、ポータル作成ポータルを利用する事で、ユーザは自分専用のポータルを即席に作成できる。我々は、PCT4Gを用いて実際にBLASTポータルを構築し、そのポータルが実用できることを確認した。

## 2. PCT4Gの概要

”Portal Construction Toolkit for the Grid (PCT4G)”は、グリッドポータルを構築するためのツールキットである。PCT4Gは、煩雑な作業なしにグリッドポータルを構築することを目指しており、複雑な機能や拡張性は提供しないが、既存のアプリケーション用のグリッドポータルを構築するために必要な作業をすべて自動化する。

グリッド構築作業を簡便にするため、PCT4Gは、既存のアプリケーションに変更を加えずに直接利用する。そのため、計算のパラメータはコマンドラインの引数、標準入力として渡される。アプリケーションのグリッド対応、並列化は行なわれないため、グリッドを利用により個々のジョブのパフォーマンスは向上しないが、複数の資源により多数のジョブが並列に実行されるため、全体的なスループットは向上する。

PCT4Gでは、グリッド構築作業に必要となるアプリケーションのインストール、アプリケーションが利用するデータ管理、インタフェースの作成を自動的に行なう。ポータル構築者はそれぞれに必要な設定ファイルを記述するだけで、グリッド構築に必要な作業はすべて完了する。そのため、従来必要であった、グリッド上の多数の計算機へのアプリケーションのインストール、設定、また、アプリケーションが利用するデータの配布、更新などの雑多な作業を行なう必要はない。アプリケーションやデータは、指定された場所からダウンロードされ自動的にインストール、配布、更新される。グリッド特有の複雑な機能は、PCT4Gに組み込まれているため、ポータル構築者はグリッド技術の深い知識なしに、グリッドポータルを構築できる。

これらを実現するため、PCT4Gではグリッド上の各資源にGlobusツールキット<sup>1)</sup>がインストールされていることを仮定する。アプリケーションのインストールやアプリケーションが利用するデータの配布、更新、ジョブの起動などはGlobusプロトコルを通じて行なわれる。そのため、PCT4G自体はグリッドポ

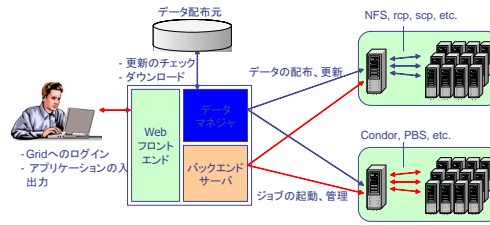


図 1 ポータル概要図

ータルとなる計算機上のみインストールすれば良く、グリッドの各資源上にはGlobusツールキット以外の特殊なソフトウェアを要求しない。

## 3. PCT4Gの設計

まず、PCT4Gで構築するグリッドポータルについて述べる。図1は、PCT4Gで作成するグリッドポータルの概要図である。作成するグリッドポータルは標準的な3層構造のシステムで、第1層はクライアントであるWebブラウザ、第2層はポータルシステム部、第3層が実際の計算を行う計算資源である。グリッドポータルの中核となる第2層は、ユーザインタフェースであるWebフロントエンド、グリッド資源の処理を行うポータルバックエンド、アプリケーションが利用するデータの管理を行うデータマネージャからなる。以下に詳細を述べる。

**Webフロントエンド** Webフロントエンドは、ユーザとのインタフェースである。Webを用いることにより、ユーザは一般的なWebブラウザでのグリッド資源の利用が可能となる。Webフロントエンドでは、ユーザのグリッドへのログインやアプリケーションとの入出力を扱う。ユーザのログインには、シングルサインオンと呼ばれる方式を用いる。シングルサインオンとは、一度の認証でグリッド上の複数の資源の使用権を確立する方式である。これにより、ユーザは使用する資源ごとに認証を行う必要がなくなる。アプリケーションとの入出力では、ユーザはアプリケーションプログラムの選択、各種パラメータの入力を行う。計算終了後は、結果がWeb上に表示される。

**ポータルバックエンド** ポータルバックエンドでは、Webインタフェースから入力された情報に基づき、グリッド上の資源にジョブを投入する。その際に、認証、データ交換、ジョブ起動、資源管理などを行う。

**データマネージャ** データマネージャでは、アプリケーションの利用するデータの管理を行なう。初回時には、データをグリッド上の各計算機上へ配布する。その後、定期的に配布元の更新をチェックし、データが更新された場合には、再びダウン

<sup>1)</sup> もちろん、適切にグリッド対応、並列化されたアプリケーションをPCT4Gを用いてグリッドポータル化する事は可能である。

ロードし配布する。

上述のグリッドポータルを構築するため、PCT4G は以下に述べるいくつかのコンポーネント、ツールより構成される。

**アプリケーションインストーラ** アプリケーションインストーラは、アプリケーションのインストールを行うコマンドラインプログラムである。ポータル構築者が記述するアプリケーションインストール設定ファイルに従いグリッドの各計算機上にアプリケーションをインストールする。

**データマネージャ** データマネージャは、アプリケーションが利用するデータの配布、更新を行うデーモンである。作成されたポータル上で動作する。ポータルバックエンド ポータルバックエンドは、作成されるグリッドポータルの内部に組み込まれ、グリッド資源の管理、計算の投入を行う。

**インタフェースジェネレータ** インタフェースジェネレータはグリッドポータルの Web インタフェースを作成する。

**グリッド作成ポータル** グリッド作成ポータルは、一般ユーザがポータルを即席に構築するための Web インタフェースである。詳細は 5 節で述べる。

#### 4. PCT4G の実装

PCT4G では、グリッドポータルとグリッド上の各サイト間の通信にはグローバルプロトコルを用い、サイト内部では各々のローカルプロトコルを用いる 2 階層の通信モデルを採用している。

グローバルプロトコルには、Globus プロトコルを用いる。Globus プロトコルは、グリッド上でデフォルトスタンダードとして使われており、これを採用することで、グリッド上の多くの資源が利用可能になる。PCT4G は、互換性、ライブラリの豊富さなどの理由から Java で実装した。そのため、Globus の Java 用のライブラリである Java CoG Kit<sup>2)</sup> を利用している。

一方、各サイト内部ではローカルプロトコルを用いる。それは、各サイトの資源がクラスタなどの複数の計算機によって構成されている場合、その代表ノードのみに Globus ツールキットがインストールをし、内部ではローカルなプロトコルを利用する運用形態が一般的であるからである。例えば、ジョブの投入には Condor<sup>3)</sup>、PBS<sup>4)</sup> 等を、ファイルの転送には NFS、scp、rcp を設定に応じて利用する。

##### 4.1 アプリケーションインストーラ

アプリケーションインストーラは、アプリケーションをグリッド資源上にインストールする。設定は、XML で記述されたアプリケーションインストール設定ファイルで行なう。設定ファイルの内容を以下に列挙する。**アプリケーションアーカイブ** アプリケーションアーカイブの場所を、GlobusURL を用いて指定する。

GlobusURL は、通常の URL の記述を拡張したもので、通常のローカルファイル (file:///...), HTTP(http://...), FTP(ftp://) の他に、Globus 標準の GSI 認証を用いたファイル転送方式である GridFTP(gsiftp:///...) を指定できる。

**インストールスクリプト** インストールスクリプトのファイル名を指定する。インストールスクリプトは、アプリケーションのインストール、設定を行うスクリプトであり、ポータル構築者が記述する。インストールする各サイトの情報 アプリケーションをインストールする各サイトの情報を記述する。各サイトの情報には、代表ノード名、サイト内で用いるファイル転送プロトコルの種類 (NFS、rcp、scp)、サイト内のノード名のリスト、アプリケーションをインストールするディレクトリ名を指定する。

アプリケーションインストーラは、まず指定された URL からアプリケーションアーカイブをダウンロードし、Globus ツールキットで採用されているファイル転送方式である GridFTP を用いグリッド上の各サイトに配布する。サイト内部では、サイト毎に、指定されたファイル転送プロトコルを用いて各ノードにアーカイブを配布する。サイト内のノードが単一の場合や、ローカルなファイル転送プロトコルに NFS を用いている場合は、各ノードへの配布は行なわない。アーカイブの配布後、インストールスクリプトを各ノード上で実行し、アプリケーションのインストールは完了する。

##### 4.2 データマネージャ

**データマネージャ** データマネージャは、アプリケーションが利用するデータの配布、更新を行う。設定は、XML で記述したデータマネージメント設定ファイルで行なう。設定ファイルの内容を以下に列挙する。

**更新頻度** データマネージャがデータの更新をチェックする間隔を時間単位で記述する。

**データファイル** 配布、更新するデータファイルの場所を、GlobusURL を用いて記述する。

**フォーマットスクリプト** フォーマットスクリプトのファイル名を指定する。フォーマットスクリプトは、データファイルの展開、フォーマットを行なうスクリプトであり、ポータル構築者が記述する。ファイルの展開やフォーマットが必要ではない場合には必要ない。

**データを配布する各サイトの情報** データを配布する各サイトの情報を記述する。指定する内容はアプリケーションインストール設定ファイルと同様である。

データマネージャは、初回はデータの配布を行なう。データの配布方法は、アプリケーションインストーラと同様である。その後は、指定された間隔で配布元のデータファイルの更新をチェックし、更新があった場

合には、再びダウンロード、配布を行う。

#### 4.3 インタフェースジェネレータ

インタフェースジェネレータでは、グリッドポータル Web インタフェースを生成する。設定ファイルは GridSpeed<sup>5)6)</sup> で採用されている Grid Application IDL で行なう。Grid Application IDL では、アプリケーション名、引数情報など Web インタフェースに必要な情報を記述する。インタフェースジェネレータは、設定ファイルを読み込み、Web インタフェースとなる JSP を生成する。実装には、GridSpeed で開発されたコンポーネントを利用している。

#### 4.4 ポータルバックエンド

ポータルバックエンドは、Web インタフェースからの入力を元にグリッド資源に対してジョブを投入する。グリッド上の資源を利用するための認証には、MyProxy<sup>7)</sup> サーバから取得した代理証明書を用いる。グリッド資源が複数登録されている場合には、まず、スケジューラにジョブを投入する資源を問い合わせ、適した資源に対してジョブの投入を行う。実際のジョブの投入には、Globus の GRAM を用いる。GRAM でのジョブ起動では、ホスト名の他に、ローカルスケジューラの種類が指定されるため、サイト内では指定されたスケジューラが利用されジョブがスケジューリングされる。

グリッド上で行われる計算の多くは長時間かかるため、バックエンドサーバはジョブの終了を待つことなく、Web フロントエンドにジョブのオブジェクトを渡す。実際にジョブが終了した際、またはエラーやタイムアウトが起こった際には、それらの情報を該当するジョブのオブジェクトに格納する。これらのジョブオブジェクトは、Web インタフェースのジョブ管理ページより参照され、表示される。

#### 5. ポータル作成ポータル

ポータル作成ポータルとは、グリッドポータルを即席に作成するためのポータルである。そのため、前節までで述べたシステム管理者をターゲットとした設定ファイルを利用する方法と異なり、Web インタフェースでの直観的なポータル構築を可能にする。(例えば、図 2 は、アプリケーションインストーラを行なう Web インタフェースである。) ユーザは、Web インタフェース上からグリッド上にログインし、ポータルに必要な情報を入力することで、設定ファイルを用いた方法と同様なポータルが構築できる。認証には、MyProxy サーバから所得される代理証明書を用いているため、アプリケーションのインストールはそのユーザが書き込み権限のあるディレクトリ(多くの場合はホームディレクトリ)に行なわれる。

このポータル作成ポータルは、自分専用のアプリケーションをグリッド上で即座に利用したい一般ユー

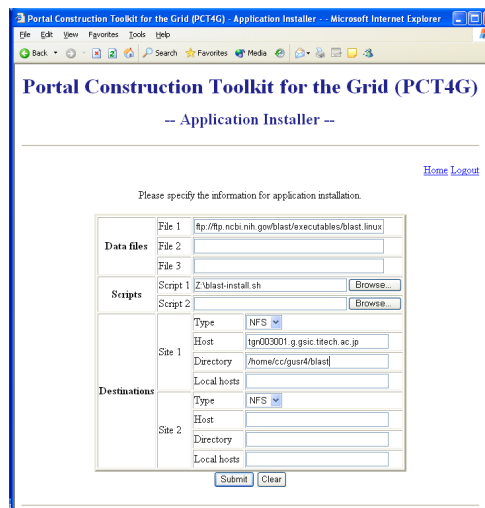


図 2 アプリケーションインストーラ画面

ザを対象としている。これらのユーザは、自分で書いた解析プログラムなどをグリッド上で実行したいユーザであり、現状では、グリッド上の利用するすべての計算機にアプリケーションのインストール、必要なデータのコピーなどを自ら行なう必要がある。グリッド作成ポータルでは、これらの作業を自動化する。

#### 6. PCT4G を用いての BLAST ポータルの構築

PCT4G を用いて、実際に BLAST アプリケーション用のグリッドポータルを構築した。BLAST (Basic Local Alignment Search Tool) はタンパク質、DNA のホモロジ検索を行うアプリケーションであり、バイオインフォマティック分野で広く用いられている。いくつかの実装が存在するが、今回は NCBI で開発、公開している NCBI BLAST を用いる。

BLAST は、検索を行なうタンパク質、DNA のデータベースを必要とする。そのデータは大きいもので約 30MB あり、毎日更新されている。既存の BLAST 用のポータルは独自にデータの更新への追従をしているが、PCT4G はデータの配布、更新の枠組を備えているためその機能を利用できる。

##### 6.1 アプリケーションのインストール

まずは、アプリケーションインストーラを用いて BLAST アプリケーションのインストールを行なった。BLAST アプリケーションのアーカイブは NCBI の FTP サイトに公開されているものを用い、インストール先には東京工業大学のグリッド資源 "Titech Grid" を利用する。

図 3 は記述したアプリケーション設定ファイルである。ダウンロードファイルは NCBI の FTP サーバ (ftp.ncbi.nih.gov) に公開されている BLAST

```

<?xml version="1.0" ?>
<data>
  <period>0</period>
  <data_files>
    <globusurl>
      ftp://ftp.ncbi.nih.gov/blast/
        executables/blast.linux.tar.Z
    </globusurl>
  </data_files>
  <scripts>
    <file>
      /home/sirasuna/blast/blast-install.sh
    </file>
  </scripts>
  <destinations>
    <site>
      <type>NFS</type>
      <host>tgn003001.g.gsic.titech.ac.jp</host>
      <dir>/home2/cc/gusr4/blast</dir>
    </site>
  </destinations>
</data>

```

図 3 アプリケーションインストール設定ファイル

のアーカイブファイル (blast.linux.tar.Z) である。これを Titech Grid 上にインストールする。Titech Grid の各ノードは、NFS によってファイルシステムを共有しているため、代表ノード (tgn003001.g.gsic.titech.ac.jp) へのみインストールする。インストールディレクトリは、“/home2/cc/gusr4/blast” である。アプリケーションをインストールするためのスクリプトには、“blast-install.sh” を用いる。ダウンロードするアーカイブはバイナリパッケージであるので、インストールスクリプトはファイルの展開を行う単純なシェルスクリプトである。

## 6.2 アプリケーションデータの配布、更新

次は、アプリケーションが利用するデータの配布と、定期的な更新の設定である。BLAST が用いるタンパク質、DNA のデータファイルを NCBI の FTP サイトよりダウンロードし、Titech Grid 上に配布、その後は更新を 24 時間に 1 回チェックし、再配布を行う設定をする。

図 4 が記述したデータマネージメント設定ファイルである。この設定では、NCBI 公開の FTP サイトに 24 時間に 1 回更新を確認し、更新があった場合には、そのデータファイルをダウンロードし、Titech Grid 上に配布する。BLAST で用いるデータファイルはフォーマットする必要があるため、そのフォーマットのコマンドを記述した“blast-update.sh”が指定されている。このシェルスクリプトは、更新があるたびに Titech Grid 上のホストで実行される。

## 6.3 インタフェース生成

最後に、Web インタフェースの生成を行なった。ア

```

<?xml version="1.0" ?>
<data>
  <period>24</period>
  <data_files>
    <globusurl>
      ftp://ftp.ncbi.nih.gov/blast/db/alu.a.Z
    </globusurl>
    <globusurl>
      ftp://ftp.ncbi.nih.gov/blast/db/alu.n.Z
    </globusurl>
    ... 略 ...
  </data_files>
  <scripts>
    <file>
      /home/sirasuna/blast/blast-update.sh
    </file>
  </scripts>
  <destinations>
    <site>
      <type>NFS</type>
      <host>tgn003001.g.gsic.titech.ac.jp</host>
      <dir>/home2/cc/gusr4/blast</dir>
    </site>
  </destinations>
</data>

```

図 4 データマネージメント設定ファイル

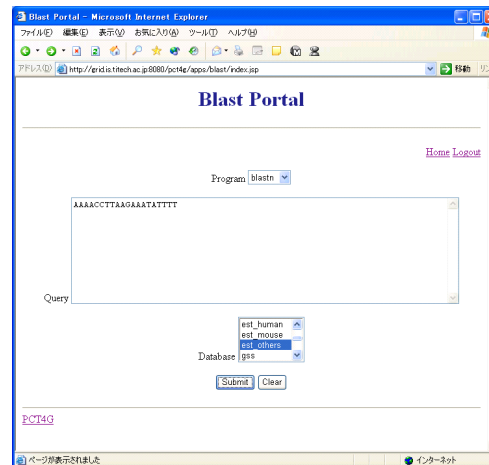


図 5 入力画面

プリケーション名、引数情報等を Grid Application IDL 用いて記述し、インタフェースジェネレータを用いて Web インタフェースを生成した。図 5 が生成した BLAST ポータルの入力画面である。

## 6.4 評価

PCT4G を用いて BLAST ポータルの構築を行なった。構築作業は、単一計算機上だけで行なえる単純なものであった。PCT4G を用いない構築方法に比べてかなり負荷が軽減されているといえる。構築した BLAST ポータルは、BLAST 専用で作り込まれた

ポータルと比較すると BLAST に特化した機能が少ないが、BLAST を利用するのに必要最小限の機能は備えている。一般的な機能として、いかに個々のアプリケーションの要望に対応していくかは今後の課題とされる。

## 7. 関連研究

グリッドポータルを構築するツールキットはいくつか開発されており、代表的なものに、NPACI の GridPort<sup>8)</sup> や LANL の Grid Portal Development Kit (GPDK)<sup>9)</sup> がある。GridPort は、Perl の CGI で記述されており、ポータル構築者は CGI を用いて、ユーザインタフェース、アプリケーションの呼出の処理を記述する必要がある。GPDK は、Java で記述されており、グリッド特有の機能は JavaBeans の形でグリッドポータル構築者に提供している。ユーザインタフェースは JSP にて記述する。これらのツールキットは、作成するグリッドポータルに多くの拡張性を与えるが、ポータル構築者がプログラミングをする必要性が生じる。PCT4G では、作成するグリッドポータルに拡張性は少ないが、プログラミングや煩雑な設定の必要なしにグリッドポータルを構築できる。

BLAST 用のポータルは、数多く存在しており、代表的なものに NCBI BLAST<sup>10)</sup> や GenomeNet によるサービス<sup>11)</sup> がある。これらは、バックエンドに高性能なサーバを用いており、BLAST プログラム自体も並列化されている。GridBLAST<sup>12)</sup> は、グリッド上で用いるように BLAST を Globus に対応されたものである。GridBLAST はコマンドラインの他、Web インタフェースを備えているため、グリッドポータルとして利用できる。PCT4G とは異なり、GridBLAST はジョブの投入時に BLAST のプログラムとタンパク質、DNA のデータを実行先のホストに転送する。その際、データは分割されて指定した複数のホストに転送される。このため、アプリケーションのインストールやデータの配布は前もって行なう必要はないが、数十 GB に及ぶデータを実行時に転送するため、パフォーマンスには優れない。

## 8. まとめと今後の課題

グリッドポータルを構築するためのツールキット (PCT4G) の開発を行った。PCT4G では、アプリケーションのインストール、データの配布、更新、インタフェースの生成をサポートし、ポータル構築の負荷を削減する。また、ポータル作成ポータルを用いることで、一般ユーザが即席に自分専用のポータルを作成することが可能である。我々は実際に PCT4G を用いて BLAST ポータルを構築し、そのポータルが実際に動作することを確認した。今後の課題は、より使いやすい Web インタフェースの作成、ジョブ管理、スケ

ジューリング、フォールトトレランスなどのグリッド特有の機能の追加、強化である。

## 謝 辞

本システムの開発は情報処理振興事業協会 (ISP) の平成 14 年度未踏ソフトウェア事業の一環として行った。

## 参 考 文 献

- 1) Foster, I. and Kesselman, C.: Globus: A Metacomputing Infrastructure Toolkit, *International Journal of Supercomputer Applications* (1997).
- 2) von Laszewski, G., Foster, I., Gawor, J. and Lane, P.: A Java Commodity Grid Kit, *Concurrency and Computation: Practice and Experience*, Vol. 13, No. 8-9, pp. 643-662 (2001).
- 3) Raman, R., Livny, M. and Solomon, M.: Matchmaking: Distributed Resource Management for High Throughput Computing, *Proceedings of the 7th International Symposium on High Performance Distributed Computing (HPDC-7)* (1998).
- 4) Altair Grid Technologies, LLC.: Portable Batch System. <http://pbs.mrj.com>.
- 5) Suzumura, T., Nakada, H., Saito, M., Matsuoka, S., Tanaka, Y. and Sekiguchi, S.: The Nif Portal: An Automatic Generation Tool for the Grid Portals, *Proceedings of the ACM 2002 Java Grande/ISCOPE Conference*, pp. 1-7 (2002).
- 6) Suzumura, T.: GridSpeed. <http://www.grid-speed.org/>.
- 7) Novotny, J., Tuecke, S. and Welch, V.: An Online Credential Repository for the Grid: MyProxy, *Proceedings of the 10th International Symposium on High Performance Distributed Computing (HPDC-10)* (2001).
- 8) Thomas, M., Mock, S., Boisseau, J., Dahan, M., Mueller, K. and Sutton, D.: The GridPort Toolkit Architecture for Building Grid Portals, *Proceedings of the 10th International Symposium on High Performance Distributed Computing (HPDC-10)* (2001).
- 9) DOE Science Grid research and development: Grid Portal Development Kit (GPDK). <http://doesciencegrid.org/projects/GPDK/>.
- 10) National Center for Biotechnology Information (NCBI): NCBI BLAST. <http://www.ncbi.nlm.nih.gov/BLAST/>.
- 11) Kyoto University Bioinformatics Center: GenomeNet. <http://blast.genome.ad.jp/>.
- 12) Krishnan, A.: GridBLAST. <http://gridblast.bii.a-star.edu.sg/>.