ビジュアルプログラミング機能を有するGridポータルシステム

福留 貴俊† 本多 弘樹† 弓場 敏嗣†
†電気通信大学

1. はじめに

近年 Grid が注目を集めている。Grid システムは広域ネットワーク上の計算資源を集合的に使用するシステムである。

Grid ポータルシステム(以下ポータル)は Web ブラウザをフロントエンドに用い、Grid サービスを利用するシステムである。これによって、ユーザは雑多なクライアントプログラムのインストールを行うことなく、Grid サービスを利用できる。Globus Toolkit[1)]等により利用基盤が整い、利用者増加が期待される今、Grid サービスへの入り口として、ポータルが果たす役割は大きくなってきている。

ポータルには大きく分けて 2 種類がある。クライアントに対して、Grid サービスを利用した Grid アプリケーションの実行環境のみを提供するものと、それに加えて、既存のリモートライブラリ等を利用した Grid アプリケーションの開発環境を提供するものである。更にその開発環境から CUI (Character User Interface)型と GUI (Graphical User Interface)型に分けられる。現在、最も利用されているのは実行環境のみを提供する Hot Page[2)]等である。スクリプト言語を用いた CUI 型開発環境を持つものには、Ninf Portal[3)]等がある。GUI 型は主にアプレットにより開発環境を提供していて、Web Flow[4)]がよく知られている。

2. 研究の目的

本研究のポータルは、GUI によるビジュアルプログラミングを可能にしている。本ポータルでは異なる Grid システム上のリモートライブラリに統一的なインタフェースでアクセスできるようにすることにより、Grid サービスをモジュール化して利用することを目指している。これにより、Grid 間のインタオペラビリティが向上し、複数のプログラミングインタフェースの利用が可能となる。

Grid portal system with visual programming
Takatoshi Fukudome†
Hiroki Honda†
Toshitsugu Yuba†
†The University of Electro-Communications

3. ポータルの設計方針

3.1 リモートライブラリのモジュール化

本ポータルでは XML を用いて、Grid アプリケーションのインタフェース情報と、既存のリモートライブラリを利用したアプリケーション本体を記述している。この XML 上では実装の異なる Grid サービスも、僅かなパラメータの違いだけで表現されている。実行時はパラメータによって、利用する Grid サービスを使い分ける。各 Grid システムの利用するサービス接続用プログラムの違い等の差は内部に隠蔽されていて、外部からは同様に扱うことができる。

統一的なインタフェースによって、複数の Grid サーバ上にある同等の計算を行うリモートライブラリとの接続を可能にするモジュールを作ることが出来る。モジュール化は以下の利点を持つ。

- 分散並列処理が可能である。
- 利用中の Grid サービスの 1 つが使用不可能になっても、計算を続行出来る。
- インタフェースが統一されているので、ポータルからのサポートがしやすい。
- GUI で簡単にモジュールを作成できる。
- ビジュアルプログラミングに適している。

3.2 モジュールを用いたプログラミング

統一的なインタフェースを持つので、本モジュールはビジュアルプログラミングに利用可能である。リモートライブラリのモジュールと同様に、統一的インタフェースを持つ変数のモジュールを用いる。これをビジュアルプログラミングのパーツに取り込むことで、様々なプログラミングインタフェースに対応できる。これにより、個々のプログラムに適したプログラミングインタフェースが選択可能である。

3.3 ポータルの構造

本ポータルはクライアント、ポータルサーバ群、Grid サービスの 3 層構造である。クライアントは Web ブラウザを用いて、ポータルサーバにアクセスする。ポータルサーバ群はデータベースを中心にして、ポータルサーバが分散配置されている。Grid サービスは本ポータルがサポートする Grid システムを用いたサービスである。

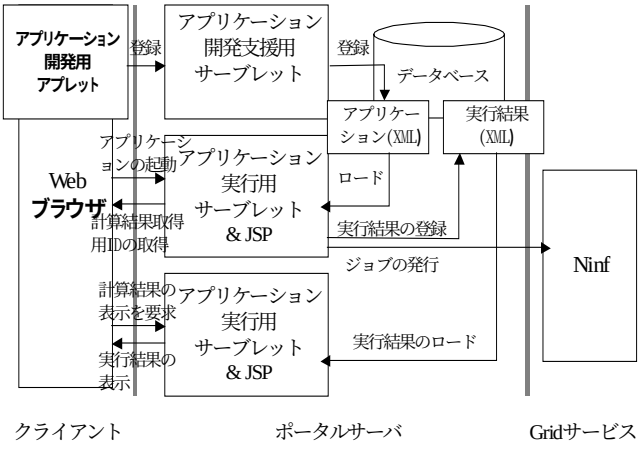

図 1 アーキテクチャ

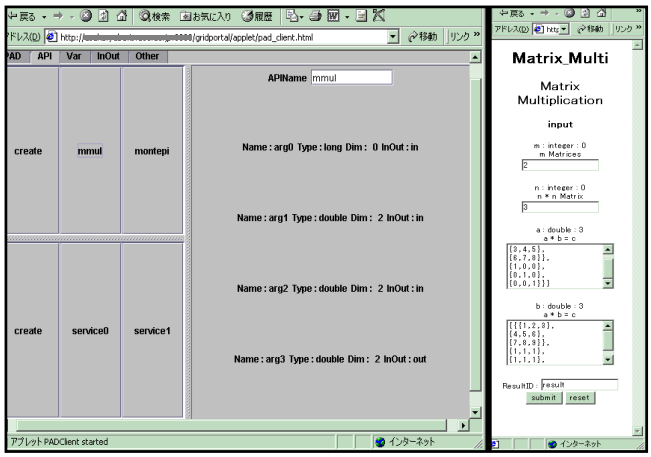

図 2 モジュールの自動生成(左), データ入力(右)

4. ポータルのアーキテクチャと実装

4.1 システムアーキテクチャ

本ポータルサーバ(図 1)はサーブレット、JSP、Java Beans を用いた MVC アーキテクチャである。複数の Grid システムに対応することを目指しているが、現在は Ninf のみに対応している。また、ビジュアルプログラミングインタフェースとしては、PAD 図の機能限定版を採用している。本ポータルの機能は開発、実行、結果表示の 3 つに分けられる。それぞれの設計について述べる。

4.2 開発環境

アプリケーションの開発時はアプレットを用いる。ビジュアルプログラミングで Grid サービスを利用するためには、Grid サービスのリモートライブラリをモジュール化する必要がある。以下のようにして、モジュールを作成する。

(1) ポータルサーバの持つインタフェース情報提供機構を用いて、ユーザの入力したパラメータから、XML で記述された Grid サービスのインタフェース情報を得る。
(2) その情報を基に Grid サービスへのインタフェースを自動生成する。
(3) 更にそのインタフェースを利用して、同様の計算を行う複数の Grid サービスを内包するモジュールの作成を行う。(図 2 左)

このようにして出来たモジュールを利用してビジュアルプログラミングを行う。完成したアプリケーションは XML 化されて、ポータルサーバへと送信される。ポータルサーバがその XML を受理したら、アプリケーションとしてデータベースに登録される。

4.3 実行手順

データベースからアプリケーションを呼び出し、実行する。

(1) アプリケーション名を入力する。
(2) アプリケーションの入力データのメタデータの入力フォームが返ってくる。
(3) データを入力し、結果の取得に必要な識別子(ID)を記述して、送信する。(図 2 右)
(4) ID が表示され、アプリケーションの実行が開始される。アプリケーションの実行はクライアントから隠蔽され、クライアントはこの時点で他の作業に移ることが出来る。
(5) アプリケーションの実行終了後、出力結果を XML 化して、データベースに登録する。

4.4 結果表示

実行時に取得した ID を入力することで、そのアプリケーションの実行結果を取得する。アプリケーションが実行中だったり、エラーが発生した場合も通知する。

5. 今後の課題

本研究の目的は一つのポータルシステムで、複数の Grid サービス、複数のビジュアルプログラミングインタフェースを持つことである。モジュール化によって、その素地は出来ているが、実際に複数の Grid サービスを持ったポータルシステムを作る必要がある。また、プログラミングインタフェースもより使いやすいものに改良する必要がある。

参 考 文 献

1) Globus Toolkit, http://www.globus.org/.
2) Hot Page, https://hotpage.npaci.edu/.
3) 鈴村 豊太郎, 中田 秀基, 松岡 聡, 関口 智嗣: 動的なアプリケーション開発実行を可能にするグリッドポータルアーキテクチャ, 情報処理学会研究報告 2002-HPC-91, pp. 191-196, 2002.
4) Erol Akarsu, Geoffrey C. Fox, Wojtek Furmanski, Tomasz Haupt: WebFlow – High-Level Programming Environment and Visual Authoring Toolkit for High Performance Distributed Computing, In Proceedings of Supercomputing '98, 1998, http://www.npac.syr.edu/users/haupt/WebFlow/.