

GridRPC システムにおけるリモートプログラム SHIPPING 機構

渡邊 啓正[†] 本多 弘樹[†] 弓場 敏嗣[†] 田中 良夫^{††}

[†]電気通信大学 ^{††}産業技術総合研究所

1. はじめに

近年、広域ネットワーク環境において、地理的に分散した多数の計算資源を容易に効率良く利用することを目的としたグリッドに関する研究が世界中で盛んに行われている。現在では、事実上標準のグリッド構築基盤ソフトウェアとして Globus Toolkit²⁾ が多くのグリッドの大学研究機関において利用されている。また Ninf, Ninf-G^{3,4)} のように、従来の RPC をグリッドに対応させた GridRPC システムに関する研究も活発に行われている。本研究では、GridRPC を利用する際にユーザ定義のモジュールを遠隔ホスト上に容易にインストールすることができる機構を開発した。

2. OpenGR コンパイラ

既存のプログラムに指示文（プラグマ）を挿入することにより GridRPC を利用できるようにする OpenGR 指示文機構とその処理系である OpenGR コンパイラが提案されている¹⁾。OpenGR コンパイラは次の処理を行う。

- OpenGR 指示文で指定された関数・部位から、GridRPC システムによって遠隔呼び出しされるモジュールを生成する。
- OpenGR 指示文で指定されたブロック内で上記のモジュールとして定義のある関数を、GridRPC システム API への呼び出しに変換する。

OpenGR コンパイラによって生成されたプログラムを実際に GridRPC システム上で動作させるためには、生成されたモジュールを GridRPC サーバ（以下計算サーバ）上にインストール（この操作をリモートプログラム SHIPPING と呼ぶ）しておく必要がある。

リモートプログラム SHIPPING は今までユーザが手作業で行ってきたが、計算サーバの台数

が多い場合やプログラムのデバッグで何回もリモートプログラム SHIPPING しなければならない場合に、ユーザの手作業による負担は大きい。本研究では、このユーザの負担を軽減し、OpenGR コンパイラによって作成された GridRPC 環境用プログラムを、容易にリモートプログラム SHIPPING できる機構を開発した。また本機構は、計算サーバ側が設定したローカル情報に従って、被呼び出しモジュールのコンパイル回数が可能な限り少なくなるリモートプログラム SHIPPING を行う。

3. 機構の設計及び実装

本研究では対象とする GridRPC 環境の構築にあたり Globus Toolkit 及び Ninf-G を用いた。本機構は次に示す機能から構成される。

3.1 リモートプログラム SHIPPING 機能

リモートプログラム SHIPPING 機能は次の処理からなる。

1. クライアント側から計算サーバ上への被呼び出しモジュールの転送
2. 計算サーバ上での被呼び出しモジュールのコンパイルと計算サーバへの登録

本機構は、ファイル転送には Globus の GridFTP、遠隔コマンド実行には Globus の GRAM を使用した Bourne シェルスクリプトで実装した。

3.2 コンパイルサーバ機能

計算サーバ群が同一のシステムの場合は、被呼び出しモジュールを特定のホスト上でのみコンパイルし、作成されたオブジェクトファイル群を他の計算サーバに転送する。

3.3 configure 自動作成機能

使用する計算サーバ群が不均質である場合、リモートプログラム SHIPPING する被呼び出しモジュールを全計算サーバ上でコンパイルできるようにするため、モジュールに対する configure スクリプトを自動的に作成する。

3.4 資源のローカル情報の授受

各資源のリモートプログラム SHIPPING に対するローカル情報は、各資源上の Globus の GRIS/GIIS に登録しておく。SHIPPING コマンド

Remote Program Shipping System on GridRPC Systems
Hiromasa Watanabe[†], Hiroki Honda[†], Toshitsugu Yuba[†]
and Yoshio Tanaka^{††}

[†]: The University of Electro-Communications

^{††}: National Institute of Advanced Industrial Science
and Technology

が実行時にユーザの権限で GRIS/GIIS からそれらを読み出して、その登録内容に従ってリモートプログラム SHIPPING を行う。

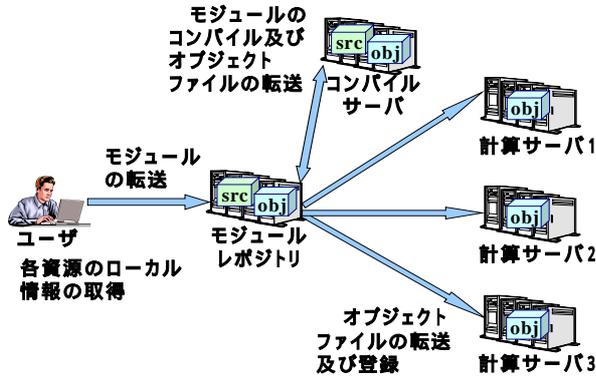


図1 リモートプログラム SHIPPING 機構

4. コマンドの利用方法及び内部処理の流れ

ユーザに要求する作業は次の通りである。各計算サーバの GRIS/GIIS には、管理者が各計算サーバのローカル情報を登録しておく必要がある。

1. 既存のプログラムに OpenGR 指示文を挿入し、OpenGR コンパイラを用いて被呼び出しモジュールを作成する。
2. リモートプログラム SHIPPING コマンドに渡す SHIPPING 情報ファイルを作成する。
3. Globus のプロキシ証明書を作成し、クライアント機上で Globus Toolkit のコマンドを利用できるようにしておく。
4. SHIPPING コマンドを起動する。

SHIPPING コマンド内部で実行される処理は次の通りである。

1. クライアント機上で、SHIPPING 情報ファイル及び各資源の GRIS/GIIS から各資源のローカル情報を取得する。
2. クライアント機上から、被呼び出しモジュールをモジュールレポジトリ上に転送する。
3. コンパイルサーバに対して、被呼び出しモジュールをコンパイルし、作成されたオブジェクトファイル群をモジュールレポジトリ上に転送するように、遠隔指示する。
4. 各計算サーバに対して、モジュールレポジトリからオブジェクトファイル群を各計算サーバ上に転送し、モジュールを各計算サーバに登録するように遠隔指示する。

5. 動作確認

使用する計算サーバ群が不均質なグリッドを再現するため、開発したリモートプログラム SHIPPING コマンドの動作確認を表1の環境で行った。動作試験は次の手順で行った。

- 電気通信大学内の1ホスト上で1回 SHIPPING コマンドを起動した。
- 電気通信大学内にモジュールレポジトリホストを1つ指定した。
- libsocket.a を必要とする被呼び出しモジュールに対し configure 自動作成機能を使用した。
- コンパイルサーバ機能を計算サーバのシステムに注意して使用した。

結果として、全ての計算サーバに対して正しく SHIPPING が行われたことを確認した。

電気通信大学
Globus Toolkit 2.0, Ninf-G 1.0, RedHat Linux 7.3 ×5台
Globus Toolkit 2.0, Ninf-G 1.0, Solaris 8 ×5台
産業技術総合研究所
Globus Toolkit 2.2, Ninf-G 1.0, RedHat Linux 7.3 ×5台

表1 動作確認に用いた計算サーバ群

6. 今後の課題

今後リモートプログラム SHIPPING 機構に実装する機能は、次の通りである。

- 計算サーバに登録されているモジュールのバージョンの隔たりを意識したりリモートプログラム SHIPPING の必要性の判別
- SHIPPING 作業の自動起動
- コンパイルサーバの隠蔽
- SHIPPING コマンドのグリッドポータルシステムへの適用
- GridRPC システム間のリモートプログラム SHIPPING 手順の規格統一による、複数の GridRPC システムへのリモートプログラム SHIPPING の対応

謝辞 本研究を進めるにあたり、数多くの御助言、御協力を頂いた、筑波大学の佐藤三久氏、産業技術総合研究所の関口智嗣氏、平野基孝氏に深く感謝致します。

参考文献

- 1) 平野基孝, 佐藤三久, 田中良夫, 関口智嗣, “OpenGR コンパイラ的设计および開発,” 情報処理学会研究報告 ハイパフォーマンスコンピューティング研究会, Vol. 2002, No.90, pp.55-60, 2002.
- 2) The Globus Project, <http://www.globus.org/>.
- 3) The Ninf Project, <http://ninf.apgrid.org/>.
- 4) 田中良夫, 中田秀基, 平野基孝, 佐藤三久, 関口智嗣, “Globus による GridRPC システムの実装と評価,” 情報処理学会研究報告 ハイパフォーマンスコンピューティング研究会, Vol. 2001, No.87, pp.165-170, 2001.