

# Globus によるジョブ分散配置システム

## The Job Distribution Arrangement System using Globus

山本 啓二  
Keiji YAMAMOTO

**Abstract:** This paper describes the job distribution arrangement system which distributes a job recursively and performs it on Grid which used Globus. That is because the program had arranged its duplicate to the remote computer, the processing which arranges a usually required program was able to be excluded. Moreover, secure communication was able to be performed between nodes by using GSI.

### 1 はじめに

近年、広域に分散された計算資源を用いて分散・並列計算を行う Grid に関する研究が盛んに行われるようになってきた。Grid は広域ネットワーク上に分散配置された計算資源を仮想的な計算機とみだてて分散・並列計算を行う計算システムである。

Grid では、ユーザ認証、セキュアな通信、遠隔計算機上でのプロセス生成など様々な技術が必要になる。Globus はこのような Grid に必要とされる基本的なサービスを提供する。Globus は Grid のための資源管理機構、ユーザ認証システム、通信ライブラリなどのツールキットを提供し、そのツールキットを用いて上位レベルに Grid を構築することが可能となる。

本稿では、Globus を使用した Grid 計算システム上でのジョブ分散配置システムの設計および構築について述べる。本システムでは単一の実行プログラムが自身の複製を Grid 上に作成し実行することで、Grid 上での再帰的なジョブの配置、起動を目標としている。

### 2 Globus の概要

Globus とは、Grid の構築を容易にするためのツールキットで、米国アルゴンヌ国立研究所と南カリフォルニア大学で共同開発されている。

Globus は Grid におけるソフトウェアを構築する際の基盤ツールとして事実上の標準になっており、世界中のほとんどすべてのグリッドプロジェクトで Globus は採用されている。

Globus はあくまで Grid を構築するツールキットであり、実態は低レベルのライブラリとコマンドの集合体である。Grid システムを構築する場合、必要に応じて Globus のライブラリを呼び出す必要がある。

### 3 ジョブ分散配置システムの設計

本節では、現在実装を行っているジョブ分散配置システムの設計について説明を行う。

#### 3.1 ジョブ分散配置システムの概略

現在実装中のジョブ分散配置システムの概略を Fig. 1 に示す。Fig. 1 ではクライアントノードと Grid 上の計算機を使用し、ジョブの起動およびノード間の P2P 接続を再帰的に行っている。

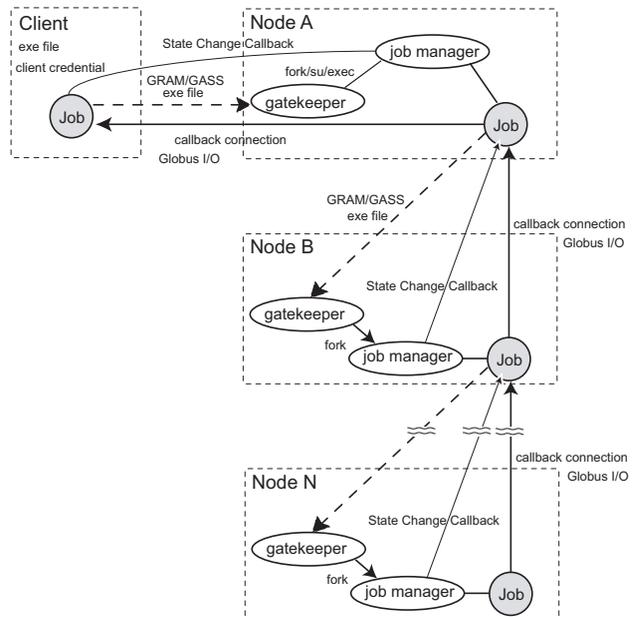


Fig. 1 ジョブの配置およびノード接続図

#### 3.2 ジョブ分散配置の手順

具体的には下記の手順でジョブを分散配置する。

step1 クライアントは自身のプログラムの複製を遠隔計算機に転送する。

step2 クライアントから GRAM API を利用して gate-

keeper にジョブ実行の要求を出す。

step3 gatekeeper がクライアントからの要求を受け取ると、クライアントと gatekeeper との間で相互認証が行なわれる。

step4 gatekeeper は適切な jobmanager を生成する。

step5 生成された jobmanager はクライアントによって転送されたプログラムを実行するためのジョブプロセスを生成する。

step6 ジョブ実行の成功/失敗/終了/結果の通知やジョブの取り消し要求などのやりとりは、クライアントと jobmanager の間で行なわれる。

step7 生成されたジョブはクライアントへ Globus I/O を用いてコールバック接続する。また、クライアントとしても動作し、別の計算機へ自身の複製を転送し、ジョブ実行の要求を出す。

step8 ジョブプロセスの出力等は Globus I/O を介してクライアントに送る。

以上の操作を再帰的に行い複数の遠隔計算機でジョブを起動する。

### 3.3 クライアント・サーバ機能

本システムでは、ジョブ起動元のプロセスとジョブ起動先のプロセスとの間に P2P のコネクションを確立する。そのためプログラムはクライアントとしての機能とサーバとしての機能を持つ。

#### 3.3.1 クライアントの機能

クライアントとしての機能は、自身のプログラムを対象となる遠隔計算機上に転送し、ジョブのキューイングを行うことである。またサーバと通信するために、ポートを listen し、サーバからのコールバック接続を待つ。

#### 3.3.2 サーバの機能

サーバとしての機能は、クライアントへコールバック接続し通信路を確立することである。

## 4 Globus による実装

### 4.1 遠隔計算機からのジョブ投入

クライアントからのジョブ起動により遠隔計算機 A で生成されたプロセスが、別の遠隔計算機 B のジョブを起動するためには、遠隔計算機 A にクライアントの権限が必要となる。Globus にはクライアントの権限を遠隔計算機に委譲する機構があり、この機構はシングルサインオンとよばれている。

具体的には、Fig. 2 に示すように grid-proxy-init で作成したクライアント証明書を遠隔計算機に送信し、遠隔

計算機はこの証明書を用いることで、クライアントの権限において別の遠隔計算機へジョブの投入を行う。

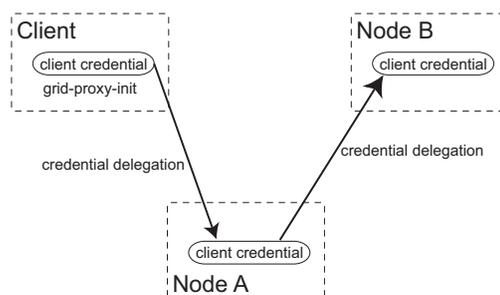


Fig. 2 クライアント権限の委譲

### 4.2 実行ファイルの配布

遠隔計算機上でプロセスを生成するには、対象となる計算機に実行ファイルが存在しなければならない。本システムでは Globus のファイル転送コンポーネントのひとつである GASS を用いて実行ファイルを遠隔計算機上に設置している。GASS は http プロトコルに GSI 認証を加えた通信プロトコルで、セキュアにデータの転送を行うことができる。

### 4.3 ジョブの起動

クライアントから GRAM API を用いて、遠隔計算機上でプロセスを生成する。クライアントは遠隔計算機上の jobmanager からジョブ実行の成功/失敗/終了/結果の通知を受け取る。また、ジョブの中断をクライアントから jobmanager に指示する。

### 4.4 ノード間の接続

GRAM により遠隔計算機上で生成されたプロセスは、ジョブ起動元のプロセスとの間に P2P な通信路を確立するため、ジョブ起動先から起動元のプロセスへとコールバック接続を行う。この通信の際に TCP/IP ソケットを拡張した Globus I/O を用いる。Globus I/O は Globus の GSI 認証や暗号化機能を備えており、それらを用いることでノード間でセキュアな通信を行うことができる。

## 5 まとめ

本稿では、Globus を用いた Grid 上において、ジョブを再帰的に分散配置し実行するジョブ分散配置システムについて述べた。プログラム自身が自分の複製を遠隔計算機に配置するので、通常必要となるプログラムの配置作業の手間を省くことができた。また、GSI を用いることでノード間でセキュアな通信を行うことができた。

### 参考文献

- 1) The Globus Project. <http://www.globus.org/>