
第1回 Globusゼミ

ゼミ担当者 : 岩橋 崇史, 釘井 睦和, 谷口 義樹
指導院生 : 片浦 哲平
開催日 : 2003年3月31日

ゼミ内容: Globus はグローバルコンピューティングのソフトウェアインフラストラクチャに必要とされる様々な要素技術を提供するツール群である。今回のゼミでは, Globus が提供するそのツール群について紹介する。

1 はじめに

1.1 Globus とは

近年, 広い地域に配置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なって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高性能計算機(メタコンピューティング)とみだてて分散/並列計算を行う計算システムである。Grid においては, ユーザ認証, 通信, 遠隔計算機上でのプロセス生成などの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。これらのツールキットは, Globus Metacomputing Toolkit (以下, Globus Toolkit) と呼ばれる。

1.2 Globus を用いた Grid に関する研究

Globus を用いた Grid に関する研究には, I-WAY や GUSTO などのテストベッドを用いた実験が行われている。I-WAY は北米の 17 のサイトに存在するスーパーコンピュータ, 記憶装置や視覚化のための装置などを ATM (非同期通信) ネットワークにより接続したテストベッドである。I-WAY 上では大規模シミュレーションなどのアプリケーションを用いて広域高性能計算の有効性を示している。また, GUSTO は北米, ハワイ, スウェーデン, ドイツに存在する全 17 サイトに置かれた 330 台の計算機(3600 台のプロセッサ)を専用の OC3 および共用のインターネットにより接続したテストベッドである。GUSTO 上では SF-Express¹を用いて, 大規模分散処理型のシミュレーションシステムを行っている。

2 Globus Toolkit

2.1 Globus Toolkit

Globus Toolkit は, オープンソースのソフトウェア・ツールキットである。プログラムのソースコードを公開し, 修正や改良を加えることができるようにしているオープンソースであるため, 世界中の多くのプロジェクトや開発者が Globus Toolkit の開発に貢献している。最新バージョンは Globus Toolkit 2.2 であり, 現在 Globus Toolkit 3.0 が開発中である(2003年1月の時点)。

2.2 提供するサービス

Globus Toolkit のサービスを Table 1 に示す。

Table 1 で示したとおり, Globus Toolkit には多数のサービスが存在する。Fig. 1 より, Globus Toolkit のサービスを利用するには, まず GSI によりユーザ認証を行う。認証を終えると, ユーザはジョブを走らせることができるサービスである GRAM や, 利用できる計算資源の情報を取得できるサービスである MDS, 遠隔の計算資源へデータ転送を行うサービスである GridFTP などを利用することができる。次章に Globus Toolkit のサービスについて具体的に紹介する。

¹Globus を用いて, 複数のスーパーコンピュータ・リソースにアクセスし, 大規模シミュレーションを行うプロジェクトのこと

Table 1 Globus Toolkit		
Service	Name	Description
Resource Management	GRAM	リソースの割り当て, プロセス生成
Communication	Nexus	Unicast/Multicast 通信サービス
Information	MDS	システムの構造, 状態に関する情報へのアクセス
Security	GSI	authentication などのセキュリティサービス
Health and status	HBM	システムの状況サービス
Remote data access	GASS	データへのリモートアクセスサービス
Executable management	GEM	実行ファイルの構築, キャッシングおよび配置

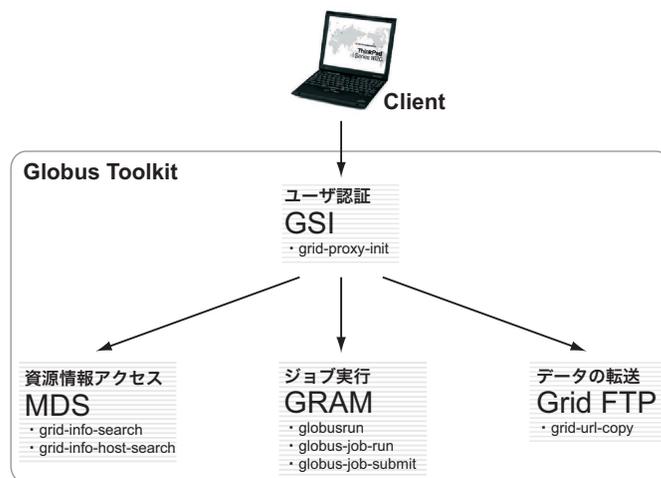


Fig. 1 Globus Toolkit の利用手順

2.2.1 GSI

GSI (Globus Security Infrastructure) は Grid におけるセキュリティサービスを提供する。GSI の認証メカニズムは、PKI (Public Key Infrastructure) と呼ばれる公開鍵暗号を用いたセキュリティ技術に基づいている。PKI とは、ユーザや計算資源などの認証単位が秘密鍵と公開鍵の対を持っており、公開鍵を流通させることによって、集中型の公開鍵の認証サーバを持たなくても任意の二者間での認証ができるセキュリティ技術である。また PKI では公開鍵を管理するために X.509 証明書を用いている。X.509 証明書は、ユーザや計算機の名前や公開鍵などの情報に対して、皆が信頼できる第三者によって電子署名したものである。信頼できる第三者として、CA (Certificate Authority, 公証局) と呼ばれる証明書の発行機関を置くことになっている。GSI では、CA は Globus が運営する Globus CA を利用することになるが、自分たちで設置することも可能である。GSI により、ユーザは計算資源を使用する際に、1 度だけ認証を行い、その時にプロセスがユーザに代わって資源を獲得できるように証明書が発行される。GSI による認証の流れを Fig. 2 と下記に示す。

1. ユーザや計算機などの認証単位が証明書を作成する。
2. CA に証明書を送り、署名してもらう。そして、署名してもらった証明書を取得する。
3. Globus Toolkit を利用したいユーザがプロキシ証明書²を発行する。
4. Gatekeeper³は、ユーザのプロキシ証明書と、すでに登録されている Mapfile⁴を照合することで認証を行う。
5. 認証後、ユーザはその計算資源で Globus Toolkit のサービスが利用可能となる。

²ユーザが Gatekeeper に対して認証を行うために発行する証明書のこと

³Client からの計算要求を受け付けるサーバプロセスのこと

⁴Gatekeeper に対して、Client が Globus Toolkit のサービスを使用できるように、Client の情報を計算機側に事前に登録したファイルのこと

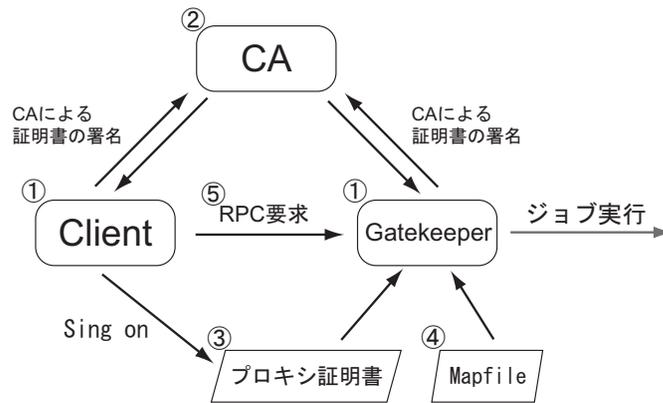


Fig. 2 GSI

下記のコマンドにより、クライアントはプロキシ証明書を発行し、Gatekeeper に対して認証を受ける。このコマンドの実行は Globus Toolkit を利用する際に必ず行わなければならない。

```

iwahashi@galley:~$ grid-proxy-init
Your identity: /O=Grid/O=Globus/OU=doshisha.ac.jp/CN=Takashi IWAHASHI
Enter GRID pass phrase for this identity: [パズフレーズ入力]
Creating proxy ..... Done
Your proxy is valid until Wed Nov 13 08:22:48 2002   プロキシ証明書の有効期限
  
```

2.2.2 MDS

MDS(Globus Meta-computing Directory Service)はGlobusのシステムを構成する、静的、動的な情報サービスへのアクセスを提供する。MDSの特徴として、どのサイトで提供されているサービスも同一の方法で得ることができ、動的データへの効率的なアクセスや複数の情報源からの同時アクセスなどを行うことが可能である。また、単一のサーバで情報を保持するのではなく、サーバが分散している非集中型管理という特徴もある。

具体的には、MDSはアーキテクチャタイプ、OS、メモリ、ネットワークバンド幅およびレイテンシ、利用可能な通信プロトコル、IPアドレスとネットワークテクノロジーとのマッピングなどの情報を保持している。問い合わせにはLDAP(Lightweight Directory Access Protocol)という情報データベース(ディレクトリ)を使用しており、MDSはFig. 3のようにGRIS(Grid Resource Information Service)とGIIS(Grid Index Information Service)により構成される。

- GRIS

個々のリソース上で実行されるサーバであり、負荷状態やプロセス情報などのリソース固有の情報を提供する。また、マシンにどれだけのメモリがあるかや大きなジョブを投入できるかなどの検索を行うことができる。

- GIIS

特定のタイプのサーバ群を管理するものであり、複数のGRISサーバから情報を収集する。マシンのうち16プロセッサ以上のSGIはどれかやホストXから100Mbps以上のバンド幅で接続できるストレージサーバはどれかなどの検索を行うことができる。更新期間の長いGRIS情報をキャッシュすることができるので、比較的静的な情報に関する組織間の問い合わせに有効である。

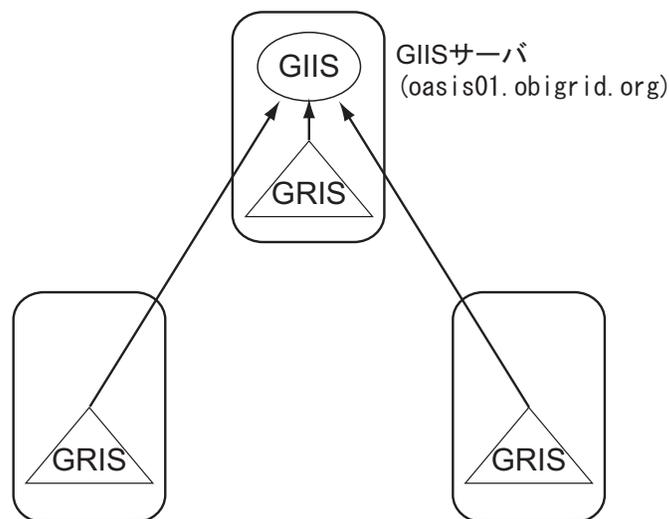


Fig. 3 GIIS

MDSを用いて使うことのできるコマンドは以下のようなものがある。

- grid-info-host-search
- grid-info-search

簡単な出力例

```
$ grid-info-search -x

-----略-----
# oasis01.obigrid.org, local, grid
-----略-----
Mds-Computer-isa: IA32
Mds-Computer-Total-nodeCount: 1
Mds-Computer-platform: i686
Mds-Cpu-Cache-12kB: 256
Mds-Cpu-Free-15minX100: 100
Mds-Cpu-Free-1minX100: 100
Mds-Cpu-Free-5minX100: 100
Mds-Cpu-Smp-size: 1
Mds-Cpu-Total-Free-15minX100: 100
Mds-Cpu-Total-Free-1minX100: 100
Mds-Cpu-Total-Free-5minX100: 100
Mds-Cpu-Total-count: 1
Mds-Cpu-features: fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pgemca cmo
  v pat pse36 mmx fxsr sse
Mds-Cpu-model: Pentium III (Coppermine)
Mds-Cpu-speedMHz: 598
Mds-Cpu-vendor: GenuineIntel
Mds-Cpu-version: 6.8.3
Mds-Fs-Total-count: 1
Mds-Fs-Total-freeMB: 11346
Mds-Fs-Total-sizeMB: 18314
Mds-Fs-freeMB: 11346
Mds-Fs-sizeMB: 18314
Mds-Host-hn: oasis01.obigrid.org
Mds-Memory-Ram-Total-freeMB: 30
Mds-Memory-Ram-Total-sizeMB: 122
Mds-Memory-Ram-freeMB: 30
Mds-Memory-Ram-sizeMB: 122
Mds-Memory-Vm-Total-freeMB: 480
Mds-Memory-Vm-Total-sizeMB: 486
Mds-Memory-Vm-freeMB: 480
Mds-Memory-Vm-sizeMB: 486
Mds-Net-Total-count: 2
Mds-Net-addr: 10.234.4.11
Mds-Net-addr: 127.0.0.1
Mds-Net-name: eth0
Mds-Net-name: lo
Mds-Net-netaddr: 10.234.4.0/24
Mds-Net-netaddr: 127.0.0.0/8
Mds-Os-name: Linux
Mds-Os-release: 2.4.19
-----略-----
```

2.2.3 GRAM

GRAM(Globus Resource Allocation Management) は計算資源(プロセッサ) 管理のためのサービスを提供する。このサービスを受けることで、ヘテロ性を考慮することなく、プログラムをリモートリソース上で実行することが可能となる。GRAMは fork, LSF や Condor など、プロセスの生成・管理の方法に応じた型(manager type)を持つ。計算サーバは1つ以上のGRAMを提供する。そして、各GRAMごとにクライアントからの計算要求を受け付けるサーバプロセスである gatekeeper をデーモンとして稼働させる。Fig. 4 に fork 型と LSF⁵型の2つのGRAMを提供する計算サーバの例を示す。Fig. 4 より、fork gatekeeper と LSF gatekeeper の2つの gatekeeper がサーバプロセスとして動作することになる。fork gatekeeper にクライアントからジョブ要求が届くと、fork 関数を使って gatekeeper が動作するホスト上でプロセスが生成されてジョブが実行される。LSF gatekeeper にクライアントからジョブ要求が届くと、LSF の bsub コマンドを使ってジョブをキューに投入し、LSF のスケジューリングに応じていずれかのホスト上でジョブが実行される。

クライアントは、リソース指定言語(Resource Specification Language : RSL) を用いて要求を伝える。実行ファイル名や、標準出力先、標準エラー出力先などを指定することができる。例を以下に示す。

```
&(count=1)(directory=/usr/local/hello)(executable=hello)(arguments=-c)
```

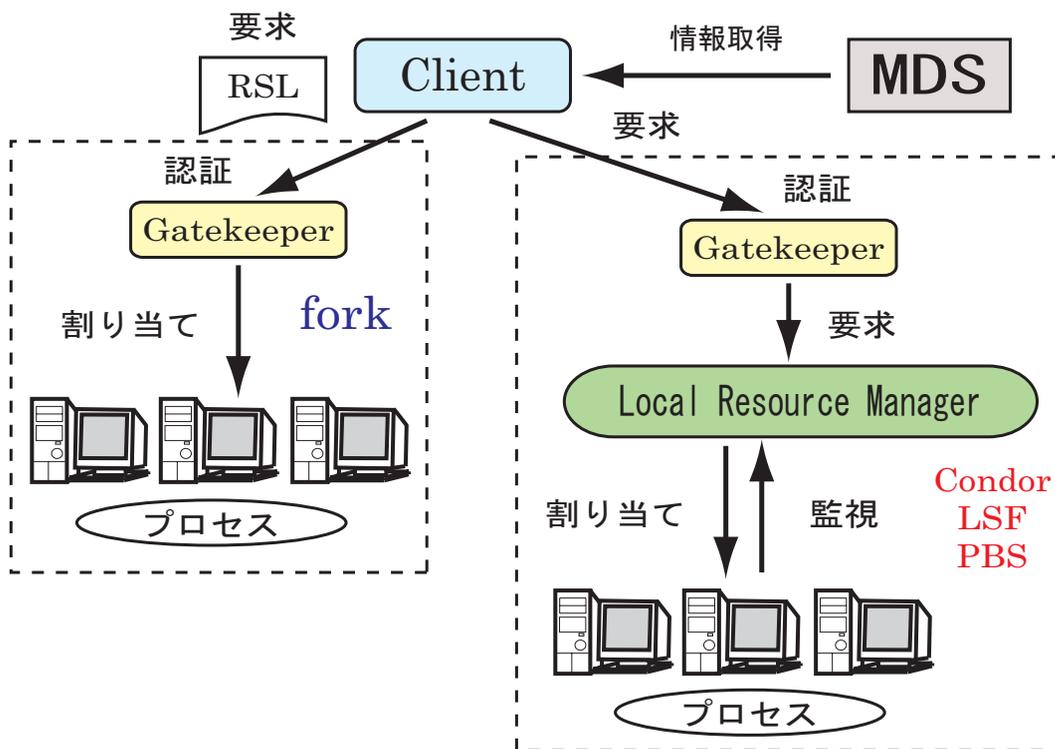


Fig. 4 GRAM

また、globus を用いて実装された MPICH である MPICH-G があるが、これは RSL を machine ファイルとして用いてジョブを実行することができる。実行コマンドは MPICH のときと同じく mpirun コマンドを用いる。

```
iwahashi@oasis02:~$ mpirun test.rsl
```

2.3 Globus の利用例

以下で、簡単な Globus のジョブを実行するコマンドを紹介する。

⁵Load Sharing Facility の略

2.3.1 globus-job-run

globus-job-run は、Foreground で動き、デフォルトでターミナルに出力が表示される。基本的には rsh(remote shell) でコマンドを実行した場合と同じである。例では、クライアント (oasis02) からリモートホスト (oasis01) に送信されたジョブが実行される様子を示している。

```
iwahashi@oasis02:~$ globus-job-run oasis01 /usr/local/globus-2.0/ga2k/ga2k
=====
Function      : Rastrigin
Chrom. Length : 100
Num Variables : 10 ( 10bits/variable )
Domain       : [ -5.12, 5.12 ) (Resolution : 0.01)

Population    : 400
Num island    : 40
Pop/island    : 10
Elite/island  : 5
init population: random
Mig. Rate     : 0.5
Mig. Interval : 5
Tournament size: 4
Selection     : Tournament
Pc            : 1
Pm           : 0.01 ( 1 / L )

Generations   : 1000
Threshold     : -1e-10
Trials        : 1
=====

trial 1 |
.....*
fitness   : 0
generation : 82
evaluate  : 33600
```

2.4 globus-job-submit

globus-job-submit は、LSF や PBS といったバッチジョブスケジューラにジョブを渡すものである。globus-job-submit はデフォルトで Background で動き、出力先はコマンドが動いているマシンである。出力を受け取りたいときには、globus-job-get-output を実行する。また、クリーンアップする場合には、globus-job-clean を、ジョブの状態を見るには、globus-job-status を実行する。

以下の例では、globus-job-submit でバッチジョブを送信する。https の URL が出力として得られる。この URL を使って、実行したジョブの状態・出力を得ることができる。

```
iwahashi@oasis02:~$ globus-job-submit oasis01 /usr/local/globus-2.0/ga2k/ga2k g10 z100 o
https://oasis01.obigrid.org:1374/14750/1039167757/
```

実行したジョブの状態を得るには、globus-job-status を利用する。状態には、PENDIN(待ち状態)、ACTIVE(実行状態)、SUSPENDED(中断)、DONE(終了)、FAILED (失敗) がある。

```
iwahashi@oasis02:~$ globus-job-status https://oasis01.obigrid.org:1374/14750/1039167757/  
ACTIVE
```

```
iwahashi@oasis02:~$ globus-job-status https://oasis01.obigrid.org:1374/14750/1039167757/  
DONE
```

globus-job-get-output で出力回収することができる .

```
iwahashi@oasis02:~$ globus-job-get-output https://oasis01.obigrid.org:1374/14750/1039167757/  
-85.3497  
-79.4934  
-70.4669  
-49.5636  
-49.5636  
-35.3405  
-32.5017  
-31.0549  
-24.7916  
-24.7916
```

2.5 globusrun

globusrun は , RSL で書かれたスクリプトを実行するものである . Foreground でも Background でも実行することができる . globus-job-run や , globus-job-submit はジョブ受渡しのために globusrun を使っている簡単なスクリプトであり , どちらも globusrun コマンドが基本となっている . 今回記述した ga2k.rsl の内容を以下に示す .

```
&(count=1)  
(directory=/usr/local/globus-2.0/ga2k)  
(executable=ga2k)  
(arguments=p150 g10 o)
```

また , 実行の様子を以下に示す .

```
iwahashi@oasis02:~/semi$ globusrun -s -r oasis01 -f ga2k.rsl
```

```
-62.7774
```

```
-51.1225
```

```
-42.5463
```

```
-23.7648
```

```
-23.7648
```

```
-20.0555
```

```
-13.49
```

```
-13.3101
```

```
-12.7561
```

```
-10.2619
```

```
=====  
Function      : Rastrigin  
Chrom. Length : 100  
Num Variables : 10 ( 10bits/variable )  
Domain        : [ -5.12, 5.12 ) (Resolution : 0.01)
```

```
Population    : 6000  
Num island    : 40  
Pop/island    : 150  
Elite/island  : 5  
init population: random  
Mig. Rate     : 0.5  
Mig. Interval : 5  
Tournament size: 4  
Selection     : Tournament  
Pc            : 1  
Pm            : 0.01 ( 1 / L )
```

```
Generations   : 10  
Threshold     : -1e-10  
Trials        : 1
```

```
=====  
trial 1 |
```

```
.  
fitness      : -10.2619  
generation   : 10  
evaluate     : 66000
```

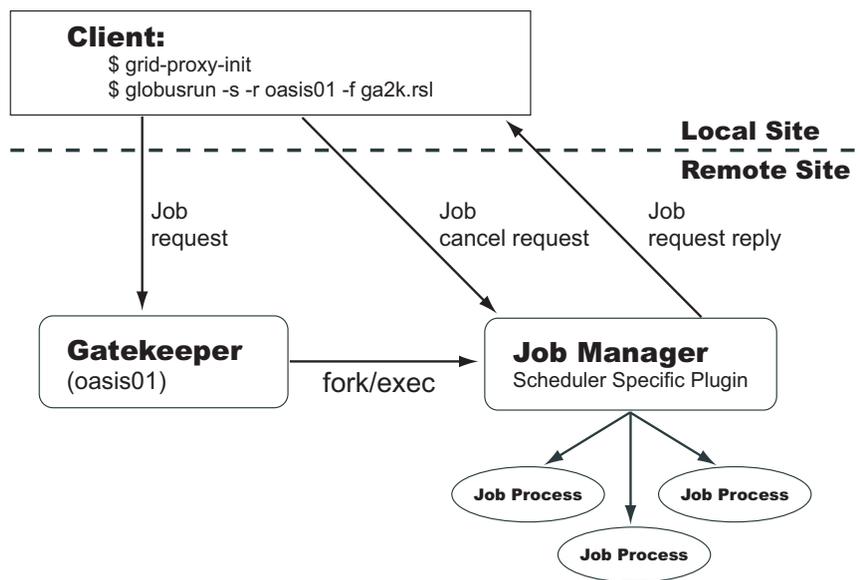


Fig. 5 globusrun コマンドによるリモートサイトでのジョブ実行の様子