

GT2を用いたソフトウェア開発

Globus Toolkit 2.x によるプログラミング

産業技術総合研究所 / 株式会社 創夢 朝生 正人



発表内容

- Globus Toolkitのプログラミングモデル
- Ninf-Gとは?
 - ▶ Grid環境でRPC機能を提供するミドルウェア <http://ninf.apgrid.org/>
 - ▶ Grid RPCの参照実装
- Ninf-G2のプログラム構成
- Globus Toolkitの問題点など



発表内容

- Globus Toolkitのプログラミングモデル
- Ninf-Gとは?
 - ▶ Grid環境でRPC機能を提供するミドルウェア <http://ninf.apgrid.org/>
 - ▶ Grid RPCの参照実装
- Ninf-G2のプログラム構成
- Globus Toolkitの問題点など



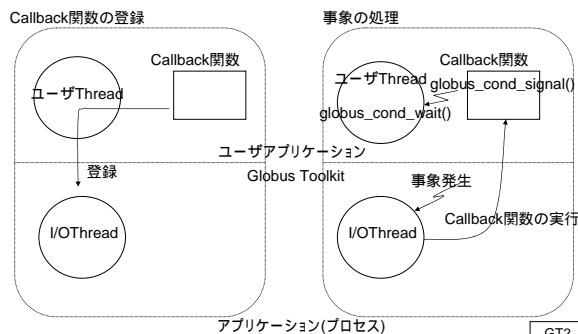
Globus Toolkitのプログラミングモデル

- Globus Thread
 - ▶ Pthread flavorとNon Thread flavor
 - ▶ Pthread flavorはPthread APIのラッパ
 - ▶ Non Thread flavorは
 - ⊕ 擬似的なThreadをエミュレート
 - ⊕ globus_cond_wait()でThreadの切り替え
 - ▶ Callback関数とMutex, Condition Variable
 - ⊕ 管理ThreadにCallback関数を登録
 - ⊕ Callback関数からglobus_cond_signal()でユーザThreadに通知

GT2



Globus Toolkitのプログラミングモデル



GT2



発表内容

- Globus Toolkitのプログラミングモデル
- Ninf-Gとは?
 - ▶ Grid環境でRPC機能を提供するミドルウェア <http://ninf.apgrid.org/>
 - ▶ Grid RPCの参照実装
- Ninf-G2のプログラム構成
- Globus Toolkitの問題点など



What is GridRPC?

GridRPC

- RPC-based programming model on the Grid
- Usage Scenario**
 - Large-scale computing on remote high performance computers
 - Large-scale matrix computation on remote supercomputers
 - Remote utilization of special purpose machine (e.g. Grape-5)
 - Task-parallel computing on cluster-of-clusters
 - Molecular simulation, etc.



GridRPC (cont'd)

v.s. MPI

- Client-server programming is suitable for task-parallel applications.
- Does not need co-allocation
- Can use private IP address resources if NAT is available (at least when using Ninf-G)
- Better fault tolerancy

Activities at the GGF GridRPC WG

- Define standard GridRPC API ; later deal with protocol
- Standardize only minimal set of features; higher-level features can be built on top
- Provide several reference implementations
 - Ninf-G, NetSolve, ...



GridRPC Model

Client Component

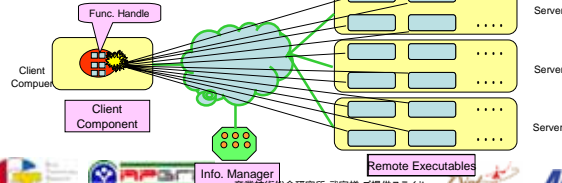
- Caller of GridRPC .
- Manages remote executables via function handles

Remote Executables

- Callee of GridRPC.
- Dynamically generated on remote servers.

Information Manager

- Manages and provides interface information for remote executables.



What is Ninf-G?

- A software package which supports programming and execution of Grid applications using GridRPC.

- The latest version is 2.2.0

- Ninf-G is developed using Globus C and Java APIs

- Uses GSI , GRAM, MDS, GASS, and Globus-I/O

- Ninf-G includes

- C/C++, Java APIs, libraries for software development
- IDL compiler for stub generation
- Shell scripts to
 - compile client program
 - build and publish remote libraries
- sample programs and manual documents



発表内容

Globus Toolkitのプログラミングモデル

Ninf-Gとは?

- Grid環境でRPC機能を提供するミドルウェア

<http://ninf.apgrid.org/>

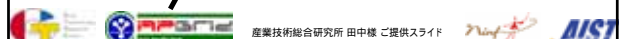
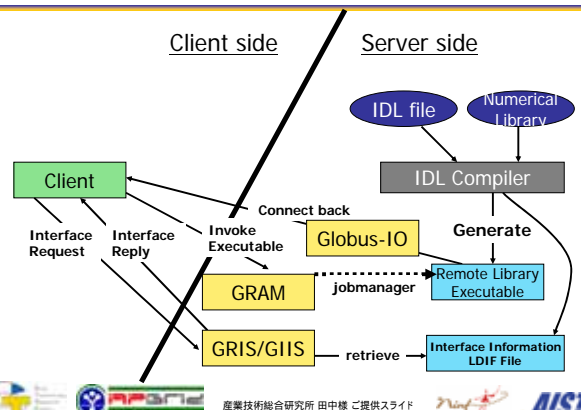
- Grid RPCの参照実装

Ninf-G2のプログラム構成

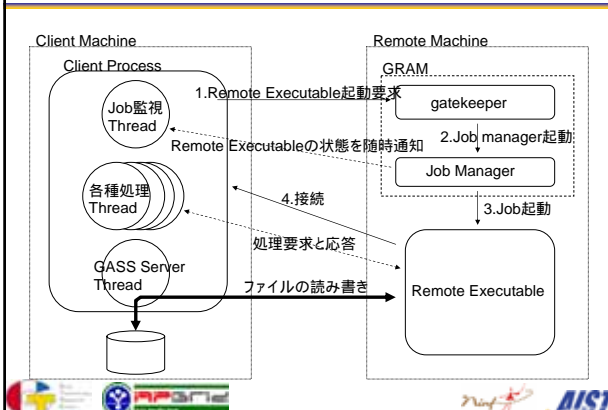
Globus Toolkitの問題点など



Architecture of Ninf-G



Ninf-Gのプログラム構成



Ninf-GのAPIとGlobus ToolkitのAPI

- `grpc_initialize()`
 - ▶ `globus_module_activate()`
- `grpc_function_handle_init()`
 - ▶ `globus_gram_client_job_request()`
- `grpc_call()`
 - ▶ `globus_io_write()`
 - ▶ `globus_io_read()`

発表内容

- Globus Toolkitのプログラミングモデル
- Ninf-Gとは?
 - ▶ Grid環境でRPC機能を提供するミドルウェア <http://ninf.apgrid.org/>
 - ▶ Grid RPCの参照実装
- Ninf-G2のプログラム構成
- Globus Toolkitの問題点など

Non Thread flavorの仕組みが不完全

- **Threadを生成できない**
 - API `globus_thread_create()`を実行すると、エラーが返る。
- **`globus_cond_timedwait()`が期待通りの動作にならない**
 - 指定時間にならなくてもすぐに戻ってしまう。

Non Thread flavorの仕組みが不完全

- **時間のコールバックがない**
 - 定期的な処理(例:ハートビート)を実行できない。
- **`globus_cond_wait()`を実行しないとGASSサーバが動かない。**
 - `globus_cond_wait()`でスレッドを切り替えるので、当然といえば当然。

Non Thread flavorの仕組みが不完全

- **Mutex, Condition VariableとGlobus I/Oの組み合わせによるデッドロック。**

```

threadA(struct object *O)
{
    globus_mutex_lock();
    while (O->inuse != 0)
        globus_cond_wait();
    O->inuse = 1;
    globus_mutex_unlock();
    :
    globus_io_write();
    :
    globus_mutex_lock();
    O->inuse = 0;
    globus_cond_signal();
    globus_mutex_unlock();
}

threadB(struct object *O)
{
    globus_mutex_lock();
    while (O->inuse != 0)
        globus_cond_wait();
    O->inuse = 1;
    globus_mutex_unlock();
    :
    globus_io_read();
    :
    globus_mutex_lock();
    O->inuse = 0;
    globus_cond_signal();
    globus_mutex_unlock();
}
    
```

Non Thread flavorの仕組みが不完全

- **Mutex, Condition VariableとGlobus I/Oの組み合わせによるデッドロック。**

```
threadA
globus_mutex_lock();
while (O->inuse != 0)
  globus_cond_wait();
O->inuse = 1;
globus_mutex_unlock();
globus_io_read();
globus_cond_wait();

threadB
globus_mutex_lock()
while (O->inuse != 0) / デッドロック /
  globus_cond_wait();
O->inuse = 1;
```

コールバック関数が呼び出される。

GT2

globus_io_try_read()の不具合

- **GSI/SSLでglobus_io_try_read()が必ず0を返す。**
- **APIのマニュアルにBugとして記述されている。**

Bug:
this function will always return 0 bytes for TCP connections which are configured to use GSSAPI or SSL data wrapping.

GT2

GRAM (GASS)とNFSの相性

- **NFS環境でRemote Jobの起動を繰り返すとJobを起動出来なくなることがある。**
 - ▶ 短時間にたくさんのJobを起動すると発生
 - ▶ GRAMが提供している複数Job起動機能を使用して回避

GT2

Globus Toolkitの性能

- **通信路にGSI/SSLの暗号化を適用すると遅い。**
- **Jobの終了に最大10秒を要する。**
 - ▶ 10秒はハードコードされている。
- **MDSが遅く不安定。**
 - ▶ 多段にするとさらに遅く不安定になる。

GT2

GT3 APIが削除された

- **いくつかのAPIが無くなった。**
 - ▶ Ian Foster氏はじめANLの数名は「意図的に消したのではない」。
 - ▶ MLで確認知ったところ開発チームは「意図的に消した」。
 - ▶ いったい誰が舵取りを行っているのか？

GT3

GT3 通信性能がGT2より極端に劣る

- **globus_io_writev()で1024件以上で極端に遅い。**
- **我々のクラスタ(Pentium III 1.4GHz, Ethernet 1Gbps)で32MByteの転送(往復)**
 - ▶ GT2:1秒
 - ▶ GT3:80秒 (1024件未満たと0.7秒)
- **GT2はwrite()繰り返し。**
- **GT3は、writev()がEINVALになった後にmmap(), memcpy(), write(), munmap()の繰り返し。**

GT3