社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2003－HPC－95 (31)
2003／8／6

次世代グリッド基盤OGSA における C 言語動作環境提供システムの試作と評価

濱野智行[†] 中田秀基[††]
鈴村豊太郎[†] 松岡聡[†,†††]

従来のグリッドアーキテクチャと Web サービスとを融合させた次世代のアーキテクチャである OGSA が注目されている。OGSI は OGSA の基盤となる技術であるが、1) 現状の OGSI 実装においては C 言語でのサービス開発が不可能、2)XML ベースプロトコル通信によるパフォーマンスの低下、といった問題が懸念されている。本稿では、OGSI 上に C/C++/Fortran の動作環境を提供するシステムを提案し、そのシステム上でサービスを開発する際の補助ツールを提供する。さらに、提案したシステムを用いて、そのシステムの有効性と現状の OGSI 実装の問題点を確認する。

Towards a C Language Hosting Environment for OGSA

Tomoyuki Hamano,[†] Hidemoto Nakada,[††]
Toyotaro Suzumura[†] and Satoshi Matsuoka[†,†††]

OGSA is a new architecture that is hubrid of traditional Grid architecture and Web services. OGSI base OGSA has two issues: 1) current OGSI implementations do not provide hosting environment for C language, and 2) XML-based protocol communication decline performance. In This paper, we propose a system that provides C hosting environment on OGSI and provide a auxiliary tool that eases developing services on the system. We also show performance evaluation results that prove effectiveness of the system and issues of OGSI implementations.

1. はじめに

近年グリッドコンピューティングと呼ばれる、広域ネットワーク上の計算資源を用いた大規模計算が注目されている。そのグリッドコンピューティングの次世代のアーキテクチャとして OGSA(Open Grid Services Architecture)[1] が提案されている。OGSA は従来のグリッドアーキテクチャと Web サービス[2] とを融合させ、他のアーキテクチャとの高い相互運用性を実現しており、また、汎用性の高いコンポーネントアーキテクチャを採用している。

OGSA の基盤である OGSI(Open Grid Services Infrastructure)[3] を科学技術計算に用いるにはいくつか問題点が考えられる。まず、現在 C 言語でサービスを記述できる OGSI 動作環境は存在しない。また、GT3 は XML ベースプロトコル通信を採用しているため、パフォーマンスの低下が懸念される。

本稿では OGSI 実装の一つである GT3(Globus Toolkit 3.0) を用いて C/C++/Fortran の OGSI 動作環境を提供するシステムを提案した。また、このシステム上でサービス開発する際の補助ツールを提供した。さらに、これらを用いて評価を行うことで、提案システムの有効性と現状の OGSI 実装の問題点を確認した。

2. 関連グリッド技術

2.1 Open Grid Services Infrastructure

OGSI(Open Grid Services Infrastructure) は OGSA(Open Grid Services Architecture) の基盤をなすものであり、Web サービスに状態を持たせることを可能にしたコンポーネントモデルを採用している。このモデルをグリッドサービスと呼ぶ。

2.1.1 OGSI コンポーネントモデル

OGSI のコンポーネントモデルを図 1 に示す。「クライアント」と「サービス」の 2 つのコンポーネントからなり、通常の Web サービスと同様に、クライアントがサービスにリクエストを送信し、その処理結果がサービスからクライアントへリプライされる。

現状の OGSI ではサービスを Java 以外の言語で記述することが不可能である。

† 東京工業大学 Tokyo Institute of Technology
†† 産業技術総合研究所 National Institute of Advanced Industrial Science and Technology
††† 国立情報学研究所 National Institute of Informatics

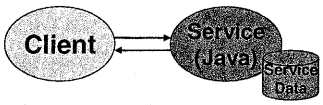


図 1 OGSI コンポーネントモデル

```
<in0 xsi:type="soapenc:Array"
 soapenc:arrayType="xsd:int[10]"
 xmlns:soapenc="http://schemas.../">
 <item>0</item>
 <item>0</item>
 <item>0</item>
 <item>0</item>
 <item>0</item>
 <item>0</item>
 <item>0</item>
 <item>0</item>
 <item>0</item>
 <item>0</item>
</in0>
```

図 2 XML エンコーディング例 (一部略)

2.1.2 サービスデータ

Web サービスとグリッドサービスとの最も大きな相違点は、Web サービスは内部に状態を保持できないのに対して、グリッドサービスはそれができるということである。この状態を外部からアクセス可能にする機構をサービスデータと呼ぶ。

サービスデータの存在によりサービスのリクエスタはサービスの問合せ、更新、通知の交換などが可能となる。

2.2 Globus Toolkit 3.0

OGSI の実装の一つに、GT3(Globus Toolkit 3.0) がある。GT3 では Apache Web サービスプロジェクト[5] によって制作されている Apache Axis[6] をベースに OGSI を実装している。さらにこの OGSI 実装を用いて GT2(Globus Toolkit 2.x) で提供されていた機能を実現している。

2.2.1 XML ベースプロトコル

GT3 では通信に XML ベースプロトコルである SOAP[7),8)] を採用している。SOAP でデータを送信する際、XML エンコーディングされた形で送信される。

XML エンコーディングする際、データサイズが一般的に巨大化するという問題が存在する。例えば要素数 10 の int 配列を XML エンコーディングすると**図 2** の様にデータサイズが大きくなる。

また、エンコーディング、デコーディングにコストがかかることも知られている。

2.2.2 サービス配置

GT3 においてサービスを開発する際に用意するコンポーネントは以下の通りである。

サービスの実装 クライアントからのリクエストを処理するサービスの本体で、現状では Java でのみ実装可能

サービスのインタフェイス記述 サービスのインタフェイスを記述した Java インタフェイス

Deployment Descriptor サービスを GT3 上に登録する際に必要

これらを GT3 に登録することで、クライアントからサービスを利用することが可能になる。

```
/**
 * The current value of the counter
 * @ogsa:service-data
 *      name = "currentValue"
 *      minOccurs = "1"
 *      maxOccurs = "1"
 *      mutability = "mutable"
 *      modifiable = "true"
 *      nillable   = "false"
 */
```

図 3 サービスデータ注釈

2.2.3 サービスデータ注釈

GT3 では**図 3** のように Javadoc コメントをメソッドの前に記述することで、自動的にそのメソッドをサービスデータの exposer に指定することができる。クライアントはその exposer を介してサービスデータを参照する。

2.3 Ninf-1

GridRPC システム Ninf-1 のコンポーネントモデルを**図 4** に示す。Ninf-1 は「クライアント」「Ninf-1 サーバ」「計算ライブラリ」からなる。

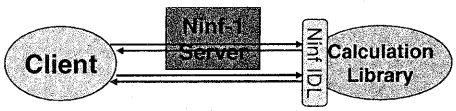


図 4 Ninf-1 コンポーネントモデル

クライアントは前もって計算ライブラリのインタフェイス情報を含む Ninf IDL を取得する。その情報を基に適切なパラメータをもって計算ライブラリをコールする。

3. システムの概要

3.1 OGSI における C 言語動作環境提供システム

提案するシステムの概要を**図 5** に示す。提案システムは「クライアント」「ブリッジサービス」「計算ライブラリ」の 3 つの要素からなる。

OGSI のサービスとして、クライアントからのリクエストをブリッジするサービスを配置し、ブリッジ先に実際にリクエストの処理を行う計算ライブラリを配置する。ブリッジサービスは Java で記述され、計算ライブラリは C/C++/Fortran で記述される。

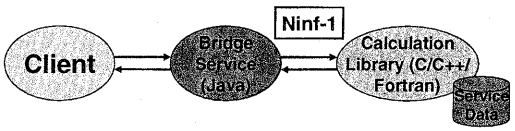


図 5　提案システムの概要図

提案システムにおいてサービス提供者が実際に用意する必要があるのは計算ライブラリである。計算ライブラリはC言語などで記述することが可能なため、提案システムを利用することでOGSI上にC言語などによるサービスを配置することが可能になる。

リクエストのブリッジにはGridRPC(Remote Procedure Call) システムNinf-1[9]を用いる。

3.2　提案システムにおけるサービス配置

提案システム上にサービスを配置する際には以下のコンポーネントを用意する必要がある。

ブリッジサービスの実装　クライアントからのリクエストを計算ライブラリへとブリッジするJavaで実装するサービス

ブリッジサービスのインタフェイス記述　インタフェイスを記述したJavaインタフェイス

Deployment Descriptor　ブリッジサービスをGT3上に登録するために必要

計算ライブラリの実装　実際にリクエストを処理するC言語などで実装されたNinf-1計算ライブラリ

計算ライブラリのインタフェイス記述　計算ライブラリのインタフェイスを記述したNinf IDL

ブリッジサービスをGT3上に登録し、計算ライブラリをNinf-1サーバに登録することでクライアントは提案システムのサービスを利用することが可能になる。

これらのコンポーネント全てをサービス提供者がサービスごとに用意するのは、非常に手間がかかる上に、その手間はサービスの本質的な部分で無いところにかかるため非効率的である。

これらのコンポーネントは多くのサービスに適用できる箇所が多く、そういった特性を利用して普遍的な部分を自動生成できる機構が望まれる。

3.3　サービスデータのブリッジ

本来サービスの状態を表すサービスデータはサービスが保持する。したがって提案システムではブリッジサービスが保持することになる。しかし、ブリッジサービスにサービスデータを保持させると、実際にサービスの処理を行う計算ライブラリからサービスデータを利用することが難しくなる。

この問題を、リクエストと同様にサービスデータもブリッジすることで解決する。すなわちサービスデータをブリッジサービスにではなく、計算ライブラリに保持させる。クライアントからサービスデータの更新・参照などのリクエストが行われると、ブリッジサービスはそれを計算ライブラリにブリッジする。リクエストを受け取った計算ライブラリは、保持しているサービスデータを用いてそれらのリクエストを適切に処理する。

4.　システムの実装

4.1　Ninf-1 on OGSI

提案システムNinf-1 on OGSIのシステム図を**図6**に示す。

Ninf-1 on OGSIは「Ninf-1ブリッジサービス」「計算ライブラリ」「クライアントアプリケーション」の3つの要素から成る。

Ninf-1ブリッジサービスは、クライアントアプリケーションからのリクエストをそのまま計算ライブラリへとブリッジする。これはNinf-1のJavaからC言語などの関数をコールする機能を用いて実装した。これによりクライアントからは透過的にC言語などで記述されたサービスを配置することが可能となった。

4.2　サービス開発補助ツール

提案システム上でサービスを提供する度に全てのコンポーネントを用意するのは非効率的であるため、それを補助するツールを用意した。そのサービス開発補助ツールNinf-1サービススケルトンジェネレータを**図8**に示す。

Ninf-1サービススケルトンジェネレータは、ブリッジサービスのインタフェイス記述(Javaインタフェイス)から提案システム上でサービスを開発する際に必要なコンポーネントのスケルトンを生成する。具体的には、以下の3つを生成する。

Ninf-1 ブリッジサービス　クライアントからのリクエストをブリッジするサービス

Ninf IDL　計算ライブラリのインタフェイス情報や実装を記述

Deployment Descriptor　サービスを配置するために必要な記述

このジェネレータはJavaのReflectionを用いて実装した。ブリッジサービスのJavaインタフェイスからクラスやメソッドなどの情報を取得し、それらを用いて上記の3つのコンポーネントを生成する。例えば**図9**のようなインタフェイスを用いると、**図10**、**図11**に示されるスケルトンが生成される。

この開発補助ツールによって提案システム上でのサービス開発者は、ブリッジサービスのインタフェイス記述と計算ライブラリを用意するだけで、C言語などで記述されるサービスの提供が可能となる。

4.3　サービスデータの実装

サービスデータを参照・更新するために必要最低限の機能を備えたメソッドをNinf-1サービススケルトンジェネレータによって生成することができる。

ブリッジサービスのインタフェイスを図9に例示す

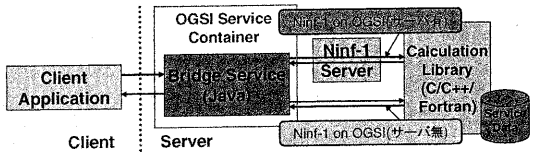


図 6 Ninf-1 on OGSI システム図

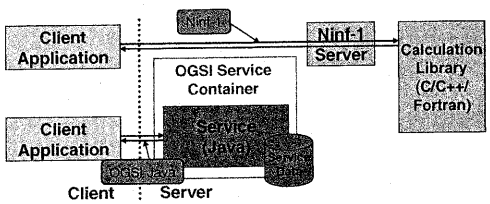


図 7 OGSI の Java 環境と Ninf-1 による RPC システム

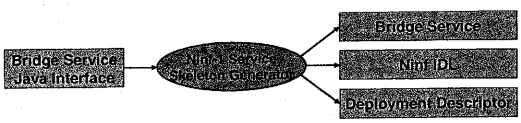


図 8 Ninf-1 サービスジェネレータ

```
package org.globus.ogsa.guide.impl;

public interface Counter {
    public int add(int val);
    public int subtract(int val);
    public void setValue(int val);
    public int getValue();
}
```

図 9 ブリッジサービスの Java インタフェイス例

るように定義すると図 11 に示す Ninf IDL が生成される。「getXXX」「setXXX」という名前のメソッドをインタフェイスに定義すると、「XXX」という名前のサービスデータが利用可能な Ninf IDL が生成され、そこに必要であれば追加・修正を加えることで容易に計算ライブラリを作成できる。

現状でサービスデータとして扱える型は、プリミティブ型、String、それらの 1 次元配列である。

5. 評　　価

提案システム Ninf-1 on OGSI を用いて、以下の評価を行った。評価環境を**図 12** に示す。

- XML ベースプロトコル通信のコストを計測するために、XML ベースプロトコルを用いる Ninf-1 on OGSI による RPC システムと、用いない GridRPC システム Ninf-1 の比較評価
- Ninf-1 on OGSI の有効性を確認するために、Ninf-1 on OGSI による RPC システムと OGSI の Java 環境による RPC システムの比較評価

```
/**
 * getValue
 * @ogsa:service-data
 *      name        = "Value"
 *      minOccurs   = "1"
 *      maxOccurs   = "1"
 *      mutability  = "mutable"
 *      modifiable  = "true"
 *      nillable    = "false"
 */
public int getValue()
              throws RemoteException {
    Vector args = new Vector();
    int[] ReturnValue = new int[1];
    args.add(ReturnValue);
    callProcedure("getValue", args);

    return ReturnValue[0];
}

public int add(int arg0)
              throws RemoteException {
    Vector args = new Vector();
    int[] ReturnValue = new int[1];
    args.add(new Integer(arg0));
    args.add(ReturnValue);
    callProcedure("add", args);

    return ReturnValue[0];
}

public void setValue(int arg0)
              throws RemoteException {
    Vector args = new Vector();
    args.add(new Integer(arg0));
    callProcedure("setValue", args);
}
```

図 10 ブリッジサービス出力例の一部

- サービスデータブリッジのコストを計測するために、ブリッジ有無による時間コストの比較評価
- Ninf-1 サービススケルトンジェネレータの定性的評価

5.1 RPC システムによる評価

提案システムを用いて RPC システムを構築し、他の RPC システムと比較評価を行った。int 型 1 次元配列を引数として受け取り、それを何も処理せず返すサービスを用いた。評価結果を**図 13** に示す。

比較評価を行った RPC システムは、図 6、**図 7** に示す以下の 4 つである。

- Ninf-1 on OGSI で構築した RPC システム (Ninf-1 サーバ有)
- Ninf-1 on OGSI で構築した RPC システム (Ninf-1 サーバ無)
- OGSI の Java 環境で構築した RPC システム
- XML ベースプロトコル通信を用いない Grid RPC システム Ninf-1

```
Module guide;

DefClass Counter
{
  DefState {
    int Value;
  }
  Define getValue(OUT int arg0[1])
  "getValue"
  {
    arg0[0] = Value;
  }

  Define add(IN int arg0, OUT int arg1[1])
  "add"
  {
  }

  Define setValue(IN int arg0)
  "setValue"
  {
    Value = arg0;
  }
  ...
}
```

図 11 NinfIDL 出力例 (一部略)

結果から XML ベースプロトコル通信を使用している RPC システムは使用していないものに比べて大きくパフォーマンスが低下していることが分かった。また、XML ベースプロトコル通信を使用している Ninf-1 on OGSI と OGSI の Java 環境とで、グラフが重なっているように、パフォーマンスに大差が無いことも分かった。

LAN 環境と WAN 環境とで結果に差がほとんど見られなかった。

5.2 サービスデータブリッジの評価

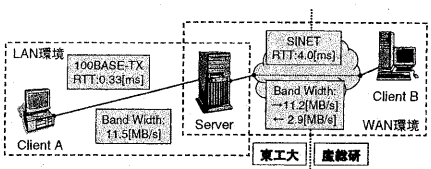
サービスデータをブリッジする際の時間コストを評価するために、サービスデータを計算ライブラリにブリッジする際の全体の転送時間と、ブリッジサービスに保持する際の全体の転送時間を比較評価した。評価結果を**図 14** に示す。

結果からサービスデータのブリッジにかかる時間コストは最大でもブリッジ無の転送時間の 9% 程度であり、ブリッジ有とブリッジ無とでほとんど差が見られないことが多かった。

RPC システムでの評価と同様に、LAN 環境と WAN 環境とで結果に大差が無かった。

5.3 Ninf-1 サービススケルトンジェネレータの評価

Ninf-1 サービススケルトンジェネレータを用いない場合、3.2 に示した 5 つのコンポーネントを用意する必要があった。しかし、このジェネレータを用いることで に示した 3 つを自動生成することができるため、結局用意すべきコンポーネントを 2 つに軽減することができた。



	CPU	メモリ
サーバ	AthlonXP 1.2GHz	1GBytes
クライアントA	Athlon 1.3GHz	1GBytes
クライアントB	PentiumIII 1.4GHz	2GBytes

図 12 評価環境

また、生成されたコードは基本的に修正を加える必要無くサービスデータを利用可能なサービス開発に使用できる。

6. 考　　察

提案システムは OGSI 上の C 言語などの動作環境として用いるに十分な性能を持ち、開発補助ツールによりサービス開発も十分容易である。だが、XML ベースプロトコル通信による性能低下のためにシステム全体としては実用性が低い。

XML ベースプロトコル通信はそのパフォーマンスの低さから、大容量データの通信を行うような科学技術計算に不適である。その様なケースに XML ベースプロトコル通信を採用する際、性能向上手法[10] の適用などが必要である。また、GT3 に XML ベースプロトコル通信のみでなく、大容量データ通信に堪えうる別プロトコルの存在は必須であると考えられる。

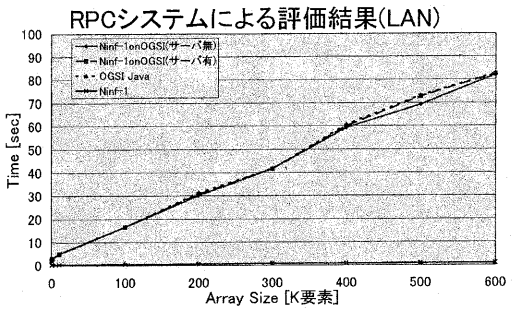
7. まとめと今後の課題

本稿では GT3 を用いて C/C++/Fortran の OGSI 動作環境を提供するシステムを試作し、その評価を行った。また、提案システム上でサービス開発する際の補助ツールを提供し、サービスデータのサポートを行った。

評価結果から提案システムは OGSI 動作環境として十分な性能を持つが、XML ベースプロトコル通信による性能低下のために実用性が低いことが分かった。

今後の課題として、以下の項目が挙げられる。

- XML ベースプロトコルの性能向上手法を用いて性能を向上させた OGSI 上での検証
- 提案システムは OGSI 動作環境としての仕様の達成
- 今後リリース予定の GT3 のネイティブな C 言語動作環境との比較評価



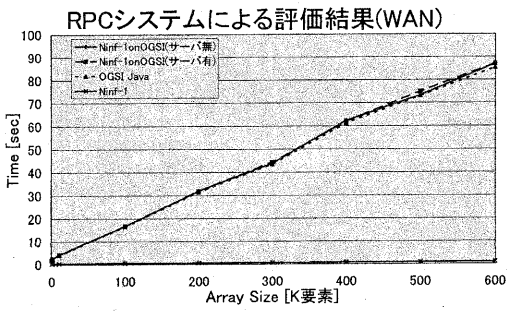
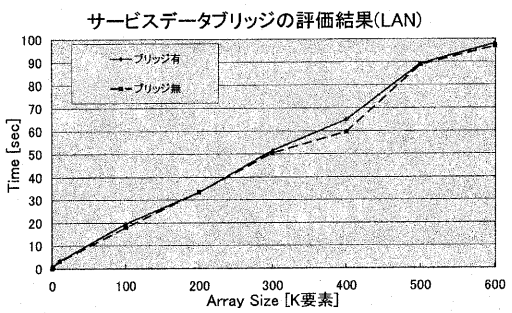


図 13　RPC システムによる評価結果



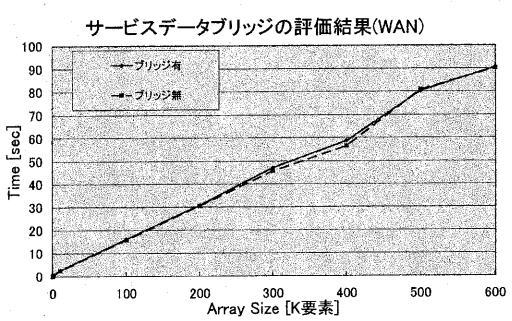


図 14　サービスデータブリッジの評価結果

参 考 文 献

1) Ian Foster, Carl Kesselman, Jeffrey M.Nick, and Steven Tuecke. *The Physiology of the Grid*, 2002.
2) *Web Services.* http://www.w3.org/2002/ws/.
3) S. Tuecke, K.Czajkowski, I. Foster, J. Frey, S. Graham, C. Kesselman, T. Maquire, T. Sandholm, D.Snelling, and P. Vanderbilt. *Open Grid Services Infrastructure (OGSI) Version 1.0*, 2003.
4) *The Globus Project.* http://www.globus.org/.
5) *Web Service Project @ Apache.* http://ws.apache.org/.
6) *Apache Axis.* http://ws.apache.org/axis/.
7) *SOAP Version 1.2 Part 1: Messaging Framework.* http://www.w3.org/TR/soap12-part1/.
8) *SOAP Version 1.2 Part 2: Adjuncts.* http://www.w3.org/TR/soap12-part2/.
9) *Ninf Project Home Page.* http://ninf.apgrid.org/.
10) Satoshi Shirasuna, Hidemoto Nakada, Satoshi Matsuoka, and Satoshi Sekiguchi. Evaluating web services based implementations of GridRPC. *HPDC11*, 2002.