

# 基于WSRF的Web服务资源的设计

易 明<sup>1,2</sup>, 金 海<sup>2</sup>

(1. 湖南文理学院计算机科学与技术系, 常德 415000; 2. 华中科技大学计算机学院, 武汉 430074)

**摘 要:** 讨论了基于 Web 服务资源框架(WSRF)的 Web 服务资源(WS-Resource)的设计。介绍了 WSRF 及其在 Globus Toolkit4 中的实现。论述了 Globus Toolkits 4 平台的基础: 公共运行时, 信息服务, 执行期管理, 数据管理, 安全性以及 WSRF 的核心: WS-Resource。结合具体步骤, 给出了利用 Globus Toolkits 4 平台设计具有良好可伸缩性的 Web 服务资源的过程。

**关键词:** Web 服务资源框架; 网格计算; 网格服务资源; Globus Toolkit 4

## Design of WSRF-based WS-Resource

YI Ming<sup>1,2</sup>, JIN Hai<sup>2</sup>

(1. Dept. of Computer Science and Technology, Hunan University of Arts and Science, Changde 415000;

2. College of Computer Science, Huazhong University of Science and Technology, Wuhan 430074)

**【Abstract】** This article focuses on the design of WSRF-based WS-Resource. It introduces WSRF(Web services resource framework) and its realization in Globus Toolkit 4. The point relates the Globus Toolkits 4 platform's foundation: common runtime, information services, execution management, data management, security, and the core of the WSRF: WS-Resource. Through the concrete step, it provides the process that constructs the WS-Resource of the good flexible function based on the Globus Toolkits 4 platform.

**【Key words】** Web services resource framework(WSRF); Grid computation; WS-Resource; Globus Toolkits 4

2002年2月, Globus项目组和IBM共同倡议了一个全新的网格标准OGSA。OGSI随后作为OGSA核心规范被提出了<sup>[1]</sup>, 其1.0版于2003年7月正式发布。但近一两年的实践证明, OGSI存在明显的不足<sup>[2]</sup>。为了解决OGSI和Web服务之间存在的矛盾, Web服务资源框架WSRF被提了出来。2004年3月, IBM、BEA与微软联合发布了WS-Addressing协议。基于该协议规范, Globus联盟和IBM迅速推出了WSRF。

Globus Toolkits 4 工具包来源于 Globus 项目, Globus 项目是国际上最有影响力的与网格计算相关的项目之一, 是由来自世界各地关注网格技术的研究人员和开发人员共同努力的成果。针对网格计算迅猛发展的现状, Globus 联盟于2005年4月29日正式发布了 Globus Toolkits 4.0.0(GT4)版, 它的特性之一就是新实现了 WSRF 和 WS-Notification 标准。所有知名的 GT3 协议都被重新设计为可以使用 WSRF, 并且 GT4 也在其中增添了一些新的 Web 服务组件。

### 1 GT4 平台体系结构

GT4 是一个开放源码的网格基础平台, 它基于开放式结构, 对用户开放服务资源和软件库组件, 并支持网格平台和网格应用, 目的是为构建网格应用提供中间件服务和程序库。GT4 平台主要用来实现以下目标<sup>[3]</sup>:

(1) 为开发一个新的基于 WSRF 的 Web 服务, 使用 Java 或 C/C++ Grid APIs 开发应用程序, 使用基础的安全机构来开发安全的应用程序等, 提供一个一致的网格开发环境。

(2) 包含一系列的基础网格服务如: 任务提交/管理服务, 文件传输服务, 数据库访问服务, 数据管理(副本、元数据)服务, 监控/索引系统信息服务等。

(3) 为用户提供一些相关的工具和例子。

(4) 为一些网格公用工具的实现提供前提。

Globus Toolkits 4 平台体系结构<sup>[3]</sup>如图1所示。

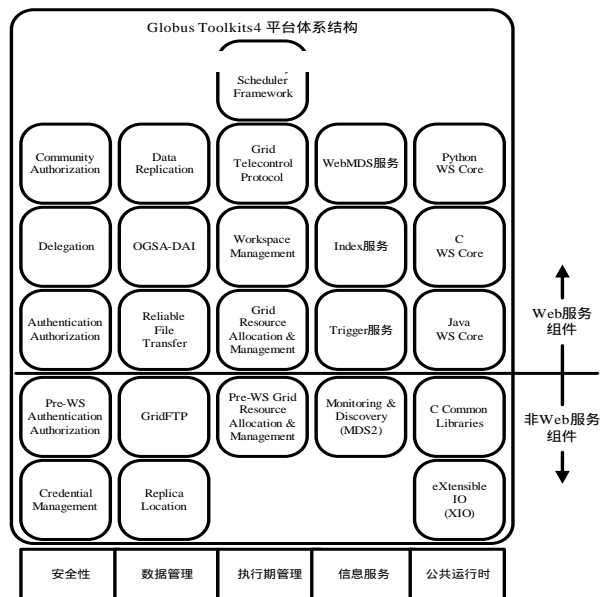


图1 Globus Toolkits 4 平台体系结构

从图1中可以看出, Globus Toolkits 4 平台体系结构共由5个部分组成, 它们分别是:

**基金项目:** 国家“973”计划基金资助项目(2003CB317003); 教育部中国教育科研网格计划“ChinaGrid”专项基金资助项目

**作者简介:** 易 明(1978-), 男, 高工、硕士, 主研方向: 网格计算; 金 海, 教授、博导

**收稿日期:** 2006-02-15 **E-mail:** hnaming@163.com

(1)Security：使通信得到安全的保证。在 GT4 中用新的基于 WS 标准的 WS Authentication Authorization 组件代替了原来的 GSI 结构，它被再细分为 Message 级安全性和 Authorizaton Framework。

(2)Data Management：允许在虚拟组织中管理大批量的数据，轻松找到及访问或移动指定的数据。GT4 中的 RLS 服务在 GT3 中的 RLS 服务基础上更加强了可伸缩性和稳定性，以及对网格环境中文件量日益增大的支持。

(3)Execution Management：涉及到作业的初始化、监控、管理、调度以及协调。GT4 提供了 GRAM 接口作为实现以上用途的基本机制。GRAM 最常用的功能就是作业提交和作业控制。GRAM 为有状态的作业控制提供了可靠的操作、异步监控和控制、远程凭证管理并实现了调度的通信协议。

(4)Information Services：一般也称为 MDS 服务，它包含一系列的组件用来在一个虚拟组织内发现和监控 WS-Resource，同时为了保持对遗留版本的兼容性，它也包含一个非 WS-Resource 版本的 MDS( MDS2)，但这个组件在以后将发布的版本中肯定会消失。

(5)Common Runtime：它是 Globus Toolkit 工具包的基础，它为开发 WS-Resource 和非 WS-Resource 程序提供了一系列的基础类库，支持 Globus 的系统服务（如 GRAM、RFT、Delegation 等）以及用户自己开发的服务。在 GT4 中，它已经被重新设计以增加对于新兴 Web 服务标准（WSRF、WS-Notification 等）的支持，同时也提供了一个使用 Java、Python 或 C 语言开发的网格服务的宿主运行环境。

## 2 WSRF 与 Globus Toolkits 4 的关系及相关规范定义

GT4 是 Globus 联盟推出的第 4 版面向网格的开发平台。第 4 版的主要特征是引入了 WS-Resource 的概念。WS-Resource 被定义为由 Web 服务和有状态资源构成的实体。有状态资源可以在 Web 服务消息交换中使用，可以创建和销毁 WS-Resource，而且可以通过消息交换查询或更改其状态。

在网格服务体系结构中有 4 个主要角色：OGSA，WSRF，GT4，Web 服务。它们之间的关系<sup>[3]</sup>如图 2 所示。

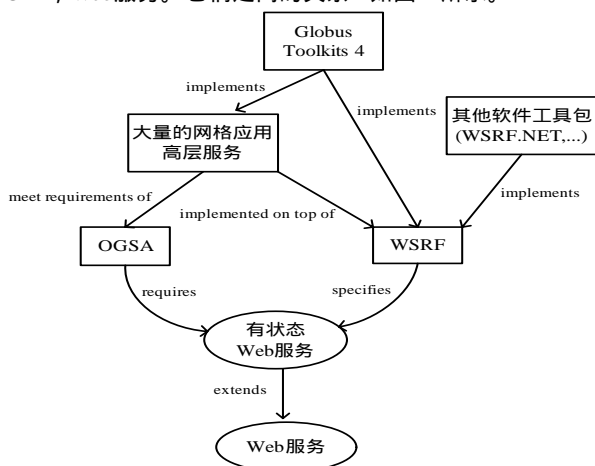


图 2 OGSA、WSRF、GT4、Web 服务四者之间的关系

WSRF 框架主要由 7 个相关的规范定义<sup>[4-6]</sup>，它们分别是：

(1)WS-Addressing：用来标准化端点引用的一个结构。端点引用给出了部署在网络端点上的 Web 服务的地址。它表示为一个 XML 序列化（XML Serialization），通常由创

建新资源的 Web 服务请求返回。

(2)WS-ResourceLifetime：定义了 WS-Resource 的资源破坏机制，其中包括允许请求方立即或使用基于时间的安排好的资源终止机制来破坏资源。它是针对 Web 服务资源的生命周期的 3 个重要方面而制订的，即 creation、identity 和 destruction。

(3)WS-ResourceProperties：定义了 WS-Resource 的类型定义如何与 Web 服务的接口描述相关联，以及如何使用 Web 服务技术来检索、更改或删除 WS-Resource 属性的消息交换过程。

(4)WS-Notification：通过基于主题的发布/订阅模式定义了事件订阅和通知机制。从 WS-Notification 角度来看，WSRF 为表示和构成通知提供了有用的构造材料；从 WSRF 的角度来看，WS-Notification 规范家族通过使请求者能够要求被异步地通报资源特性值的改变来扩展 Web 服务资源的效用。

(5)WS-RenewableReferences：定义了端点变为无效时，对需要检索端点引用更新版本的策略信息的常规 WS-Addressing 端点引用。它定义了一些可应用于任何的端点引用的机制用来 renew 已经无效的端点引用。

(6)WS-ServiceGroup：定义了一种方法，使 Web 服务和 Web 服务资源可以为了某个领域的特定目的而聚集或组合在一起。为了让请求者能够根据 ServiceGroup 的内容进行有意义地查询，可以通过使用分类机制以级别来对组中的成员资格进行限制。

(7)WS-BaseFaults：为在 Web 服务消息交换过程中返回的错误信息定义了基本的 XML 错误类型以及 Web 服务如何使用这种故障类型的规则。

## 3 基于 WSRF 的 WS-Resource 开发

WSRF 采用了与网格服务完全不同的定义：资源是有状态的，服务是无状态的。WSRF 推出的目的就在于定义出一个通用且开放的架构，利用 Web 服务对具有状态属性的资源进行存取，并包含描述状态属性的机制，另外也包含如何将机制延伸至 Web 服务中的方式。因此，具有状态属性的 Web 服务资源的开发是基于 WSRF 的网格应用程序开发的关键，下面以 Web 服务资源 CashJournal 为例，描述在 GT4 平台下开发一个 Web 服务资源的基本模式。CashJournal 服务的主要功能是：根据客户端传进的参数信息，对个人账户余额进行相应数额的加减，并回显个人账户当前余额。CashJournal 服务共包括 Client、ResourceHome、Web service、Resource 4 个部分，各部分之间的关系如图 3 所示。

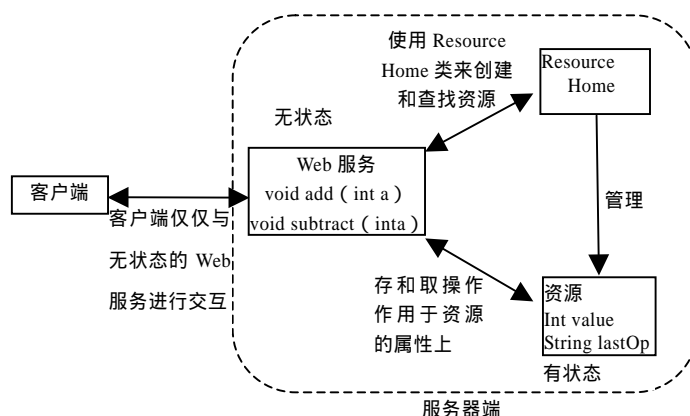


图 3 Service、Resource Home、Resource、Client 四者之间的关系

### 3.1 程序结构

(1)环境变量：在真正开始创建 CashJournal 应用实例前，还需要先设置以下 3 个系统环境变量，然后下载 Junit 3.8.1 安装包，拷贝 junit.jar 文件到\$ANT\_HOME/lib 目录中。

1)JAVA\_HOME=/usr/local/jdk1.5 (JDK 安装路径)；

2)ANT\_HOME=/usr/local/ant\_1.6 (ANT 安装路径)；

3)PATH=\$ANT\_HOME/BIN:\$JAVA\_HOME/BIN:\$PATH  
(命令搜索路径)。

(2)服务接口：为了将服务描述清晰、明了，CashJournal 服务仅实现了 3 个基本的操作：存入，支出和取余额。如果能实现这 3 个最基本的操作，要实现其它更复杂的操作仅是再综合调用这 3 个操作而已。

```
public interface CashJournal{
    public void add(int a);
    public void subtract(int a);
    public int getValueRP();
}
```

(3)服务及资源的实现：在得到能够准确描述 Web 服务的接口文件后，还只能说 CashJournal 服务能进行账户的管理了。即只是已经知道它能做什么了，但还没有指定它怎么做。实现服务和资源的详细步骤如下：

1)定义 QNames 接口：当客户必须引用与这个服务相关的所有事物时，需要使用 qualified names (QNames)，它包含了一个 namespace 和局部名称。一个 Qnames 在 Java 中用 QName 类来进行描述。

2)实现有状态资源：有状态资源的实现类包含了实现这个资源以及对它的属性进行操作的所有接口代码，这些都是有状态的组件。

3)实现有状态资源的 Resource Home 类：实现 Resource Home 类是非常简单的，因为可以从 GT4 包括的 Singleton Resource Home 类继承实现。这个基类几乎提供了所有 Resource Home 类用来管理单个资源的功能，只需要实现 findSingleton 方法就行了。当用户第 1 次请求资源时，这个方法将会被 ResourceContext 类调用。findSingleton 方法可能建立一个新的资源对象、初始化它、返回它。Singleton Resource Home 类将保留资源的一个副本，以后每次当资源被请求时，就将其返回。

4)实现无状态 Web 服务：接下来准备实现 Web 服务。Web 服务是用满足一定要求的 Java 类来实现的。在这个类中将为 Web 服务接口 (CashJournal.java) 提供声明。这里以其中一个具体操作：ADD ( ) 操作为例进行说明。

```
public AddResponse add ( int a ) throws RemoteException{
    //通过 getResource ( ) 方法获得对一个资源的引用
    CashResource cashResource = getResource ( );
    //对资源上的有状态信息进行各种操作
    CashResource.setValue ( cashResource.getValue ( ) +a );
    cashResource.setLastOp ( "ADDITION" );
    return new AddResponse ( );
}
```

### 3.2 部署服务与资源

到现在为止，Web 服务和资源所需要的服务接口、Qnames 文件、WSDL 文件、Resource 文件、Resource Home 文件和服务实现的全部 Java 代码已经编写完毕。接下来需要编译所有的代码并把它们部署进 Web 服务器 (在这里是 the

GT4 standalone container)。部署服务与资源的具体实现过程如下：

(1)创建 Globus Archive 包 (GAR 包)：首先需要创建称为 GAR 的特殊 JAR。这个才是部署进 Web 服务器的实际文件。把所有东西整齐地打包进一个 GAR 包可以让服务和资源的部署更容易，并很容易将这个 Web 服务资源包在另一个不同的计算机上部署。

由于事先已安装好了 Python 软件，因此现在创建 GAR 包和部署 GAR 包的过程都相对简单很多。只需执行 globus-build-service.py CashJournal 命令，该命令执行完毕将在当前目录下产生一个名叫 org\_globus\_examples\_services\_core\_CashJournal.gar 文件。

(2)部署 GAR 包：在上一步创建 GAR 文件后，就可继续利用 Python 软件来简化部署的过程。执行 globus-deploy-gar org\_globus\_examples\_services\_core\_CashJournal.gar 命令，部署完毕。

### 3.3 功能测试

(1)启动服务器端容器程序

Globus-start-container -nosed

(2)对 Web 服务资源进行访问

java org.globus.examples.clients.CashJournal\_instance.Client

http://192.168.1.99:8080/wsrf/services/examples/core/CashJournal  
0100

//0 代表操作方式存(1 为取)，100 代表操作金额数目

### 3.4 注意事项

在启动服务器端容器程序和编译、运行客户端程序前，都必须先执行一次 globus-devel-env 批处理文件，以设置相应的类库路径。如是在单机环境下进行模拟实验，则可通过使用“开始”菜单中的“运行”项，建立两个 DOS 窗口。

### 4 结束语

Globus Toolkit 4.0.0 通过引入 WSRF 规范，使程序员可以方便地声明和实现 Web 服务和一个或者多个有状态的资源之间的关联，从而设计出更灵活、更商业化的 WS-Resource 应用程序，这是对传统网格开发模式提出的挑战。理解 GT4 平台的体系结构模型是网格服务开发的基础，遵循 GT4 网格服务开发的基本模式，是网格服务程序设计的逻辑指导。在网格应用研究中，GT4 已经成为建造网格环境的首选工具。

### 参考文献

- 1 王 毅, 李智深, 刘 鹏. Globus: 网格世界的 Linux[J]. 中国计算机用户, 2005, (8).
- 2 邹德清, 金 海. 网格服务体系结构的演变[J]. 中国计算机用户, 2005, (2).
- 3 Borja S. The Globus Toolkit 4 Programmer's Tutorial[Z]. [http://gdp.globus.org/Globus Toolkits 4-tutorial/singlehtml/progtutorial\\_0.1.html](http://gdp.globus.org/Globus%20Toolkits%204-tutorial/singlehtml/progtutorial_0.1.html).
- 4 应 宏, 王自全, 陈晓峰. 网格与 Web 服务的融合——OGSA 与 WSRF[J]. 重庆三峡学院学报, 2005, 21(3): 41-44.
- 5 孙 炜, 任长明, 朱 江等. 基于 Web Service 的网格资源架构 WSRF[J]. 微处理机, 2005, 26(1): 13-16.
- 6 刘会斌, 都志辉. 网格与 Web 服务的融合——WSRF 和 WS-Notification[J]. 计算机科学, 2005, 32(2): 76-79.