# 基于OGSA的网格编程技术

李 季[1,3] 黄小勇[2] 张 伟[3]

(重庆大学计算机学院 重庆400044)[1] (重庆大学工商管理学院 重庆400044)[2]

(重庆教育学院计算机现代教育系 重庆400067)[3]

**摘 要** 目前,网格技术作为一种新的计算范式正在兴起。网格论坛组织提出的OGSA正成为网格应用的通用和标准化的体系结构,其核心就是网格服务,它是Web Service和网格技术的结合。本文对该体系结构进行了概述,并对网格程序的编写模式进行了归纳和总结。

**关键词** OGSA,网格服务,网格

## A Programming Technology Based on OGSA in Grid Environment

LI Ji HUANG Xiao-Yong ZHANG Wei

(College of Computer Science,Chongqing University,Chongqing 400044)[1]

(College of Business Administration,Chongqing University,Chongqing 400044)[2]

(Dept. of Computer&Modern Education Technology,Chongqing Education College,Chongqing 400067)[3]

**Abstract** As a new paradigm,Grid Computing (Computational Grid ) is springing up. OGSA(Open Grid Services Architecture) proposed by Global Grid Forum is becoming a common and standard architecture for grid-based applications. Grid service formed from Web service and grid technic is the key issue. This paper describes the architecture in short,and then followed with programming technology based on OGSA.

**Keywords** OGSA,Grid service,Grid

## 1 引言

随着Internet技术的发展和普及,建立在其上的网格计算越来越受到人们的重视和关注。简单说来,网格计算就是下一代的分布式计算,其目标就是将地理分布的各种异构资源通过Internet连接,创建成一个简单但规模和性能巨大的自组织的虚拟的计算机[1]。网格从类型上可以分为:计算网格、数据网格和信息网格等。不管网格如何进行分类,现在其基本编程模式多数都是遵从OGSA(Open Grid Services Architecture)标准。OGSA是面向服务的体系结构,在OGSA中,一切硬件和软件都称之为服务。它是Web Services和Grid技术融合的产物,遵循Web Service标准,并对之进行了扩展。

因此,本文首先对OGSA进行简单的概述,然后对按照OGSA体系结构进行编程的技术进行介绍。

## 2 开放式Grid服务体系结构(OGSA)

Open Grid Services Architecture (OGSA)的目标是为基于网格的应用定义一个通用标准的体系结构,它是网格论坛组织提出的一种网格体系结构。OGSA是集成关键的网格技术和Web Service机制来创建基于开发式Grid服务平台的分布式系统框架[2],它利用和Web Services同样的底层结构,例如:XML、SOAP、WSDL,以及WSIL等。但动态网格环境必须具有对分布式的原子或集合服务的可控的、可容错的和安全的管理机制来创建和发现定制服务实例,因此,它对Web service进行了许多重要的概念上和应用上的扩展来满足这样的要求。

OGSA定义了所谓的Grid服务:提供一套良好定义接口和遵循一定规范的Web服务。OGSA的一个基本前提是任何事务都表示成一个服务:一个网络可达的、通过消息交换提供某些功能的实体。计算资源、存储资源、网络、程序、数据库等都是服务。采用一个统一的面向服务的模型意味着环境中的所有组件都是虚拟的。接口解决发现、动态服务创建、生命周期管理、通知和可管理性等问题;规范解决命名和可升级能力的问题。

一个Grid服务可以实现一个或多个接口,每个接口定义一个操作的集合,调用操作通过一个定义好的消息交换序列来实现。Grid服务接口对应于WSDL中的portType(端口类型)。一个Grid服务所支持的portType的集合和一些附加的版本信息在该Grid服务的serviceType中指定,serviceType是OGSA定义的一个WSDL可扩展元素。

尽管OGSA定义了多种行为和相关接口,除了一个接口(GridService)外,其他所有接口都是可选的。该端口负责查询有关Grid服务实例的多种信息;设置(和获取)Grid服务实例的终止时间;终止Grid服务实例。

网格服务具有四个最基本的功能[2]:动态服务创建、服务发现、服务生命周期管理、通知。下面就分别对此进行介绍。

### 2.1 动态服务创建

动态地创建和管理新的服务实例的能力是OGSA模型的基本原则,使得服务创建服务(service creation services)的存在成为必要。OGSA模型定义了任何服务创建服务都必须提供的一个标准接口(Factory)和相关语义。OGSA定义了一类Grid服务,实现一个创建新的Grid服务实例的接口。该接

李 季 博士生,主要研究方向为网格计算、分布式计算机网络。

口称为 Factory，把一个实现了该接口的服务称为一个工厂(factory)。Factory 接口的 CreateService 操作创建一个请求的 Grid 服务，并返回 GSH 和新的服务实例的初始 GSR。

OGSA 服务可以动态创建和销毁。服务和服务实例类似于面向对象中的类与对象的关系，不同的服务实例具有不同的内部状态，服务实例可以被显示地销毁，或由于某些系统故障，例如操作系统崩溃或网络隔离而被销毁或变得不可达。

因为 Grid 服务是动态的和有状态的，所以需要一种方法来区分一个动态创建的服务实例和另一个动态创建的服务实例。因此，每个 Grid 服务实例被分配一个全局的唯一命名：Grid 服务句柄(Grid Service Handle，GSH)，以将一个特定的 Grid 服务实例区分于所有其他的 Grid 服务实例，包括已存在的、现在存在的或将来会存在的[4](如果一个 Grid 服务失败了，以同样的方式重启并保持原来的状态，则其本质上是同一实例，使用同样的 GSH)。

GSH 不携带任何特定于协议或特定于实例的信息，例如网络地址、支持的协议绑定等。取而代之的是，这些信息和所有其他与某个特殊服务实例交互所需的特定于实例的信息被封装在一个叫做 Grid 服务引用(Grid Service Reference，GSR)的单一抽象中。GSR 不同于 GSH，GSH 是 URI 的一个子集，它是常量，它本身并不携带可以直接访问网格服务实例的信息。客户端要同服务实例进行通信，必须将 GSH 解析为 GSR，GSR 包含客户端与服务实例通信所必须的信息。

一个 Grid 服务实例的 GSR 可能会在服务生命周期内发生改变。一个 GSR 有一个显式的终止时间，或随时会在一个服务的生命周期内变得无效，OGSA 定义了获得一个最新的 GSR 的映射机制。

任何分布式系统都必须能够处理不可避免的失败。在一个由短暂的、有状态的服务实例组成的系统内，必须提供回收服务机制和与失败操作相关的状态机制。通过定义两个标准操作解决了这一需求：Destroy 和 SetTerminationTime(在必需的 GridService 接口内)，分别用于显式地销毁 Grid 服务实例和 Grid 服务实例的软状态生命周期管理(软状态协议允许在远程创建状态，并最终抛弃该状态，除非被一系列随后的“keepalive”消息进行刷新。这样的协议具有易于失败恢复的优点：一个丢失的消息不会导致不可挽回的损害、和简单的优点：无需可靠的“丢弃”协议消息)。

### 2.2 生命周期管理

短暂服务实例引发了确定服务生命周期的问题：即确定一个服务什么时候可以或应该被终止，以便回收相关资源。在正常的操作条件下，一个短暂服务实例被创建以执行一个特定的任务并在任务完成时终止或通过请求者或请求者指派的另一服务的显示调用来终止。然而，在分布式系统中，组件有可能失效，消息有可能丢失。由此而产生的一个后果是服务有可能永远也不会收到一个期望的显示的终止请求，因而导致服务无限地消耗资源。

OGSA 通过一个软状态协议解决上述问题[3]，在该协议中，Grid 服务实例在创建时会具有特定的生命周期。通过客户端或代表客户端(当然，遵循服务的策略)的另一 Grid 服务的显式请求，可以将初始生命周期延长到一个指定的时间。如果到了该指定时间而未收到客户端的重新确认，则宿主环境或服务实例本身有权终止服务实例，并释放所有相关资源。

### 2.3 服务发现和服务数据

服务发现是通过 GridService 接口的 FindServiceData 操作来实现的。服务发现的本质是获得所请求的服务的 GSH。为此，Grid 服务规范为每个 Grid 服务接口定义了一个、零个或多个称之为服务数据元素的集合，该元素集合包含有关一个 Grid 服务实例的基本信息，包括它的 GSH、GSR、主密钥和主 handleMap 等。

一个支持服务发现的 Grid 服务称为一个注册点(registry)。一个注册服务由两部分进行定义：第一部分是信息部分，封装有关注册的 GSH 信息的相关服务数据元素；另一部分是 Reigstry 接口，该接口允许在一个注册服务上注册一个 GSH，并将其加入到属于同一子集的 GSH 集合内。一个服务经过注册后，其它服务就可以通过 GridService 接口的 FindServiceData 操作来获取已注册的 GSH 的信息。

### 2.4 通知

一个动态的、分布式服务的集合必须能够异步地相互通报有意义的状态改变。OGSA 通知框架允许客户端注册(订阅)感兴趣的特别消息(NotificationSource 接口)，一旦产生该消息，就通知该消息的客户。OGSA 为订阅(NotificationSource)和传输(NotificationSink)这些通知定义了通用的抽象概念和服务接口，以便由更简单的服务组合而成的服务能够以标准的方式处理这些通知(例如出错通知)。NotificationSource 接口与服务数据相整合，以便一个通知请求可以表示成对服务数据的连续“推”模式传输的请求。

该框架支持异步的、单向传输通知(NotificationSink)。如果一个特殊的服务想要支持通知消息的订阅，则它必须支持 NotificationSource 接口以管理订阅。一个希望接收通知消息的服务必须实现 NotificationSink 接口，该接口用于发送通知消息。为了从一个特殊服务开始通知操作，我们调用通知源接口上的订阅操作，返回通知接收器的服务 GSH。接着，一个通知消息流从源流到接收器；同时，接收器发送定期 keepalive 消息以通知源它仍有意于接收通知。

## 3 利用 Globus Toolkit 3(GT3)开发网格服务

GT3是遵循 OGSA 标准的网格开发工具包，当前，它已成为众多网格项目开发的主流工具。Globus 是美国 Argonne 国家实验室的项目，初始阶段全美有10多所大学和研究机构参与。目前的主要参与者有：南加州大学的信息科学学院、芝加哥大学、爱丁堡大学、瑞典的并行计算机中心等。Globus 联盟1999年推出 Globus Toolkit 的第一个版本，其后在2002年初推出2.0版，年底推出2.2版(IBM 红皮书)。2003年推出 GT3.0 的 alpha，beta，以及正式版。Globus Toolkit 是一个软件工具包，它允许编写基于 OGSI 网格服务的应用。

网格应用程序的开发，其实就是对服务的编写。简单服务可以组合成复杂的服务。下面对一个服务的编写进行总结介绍。

利用 GT3工具包对网格服务开发必须经过下列服务开发和部署的基本步骤[5]：①提供一个服务接口；②生成 Stub 文件；③实现服务；④创建部署文件；⑤部署服务。

### 3.1 提供一个服务接口

编写网格服务，首先就是确定服务的功能。服务通过服务接口向外界提供操作，服务接口在 Web Service 术语中被称为端口类型(通常被写为 PortType)。网格服务提供的操作通过 XML 语言的 Web Service 描述语言(WSDL)进行指定，编写服务接口有如下两种方式：

1. 用 java 语言编写一个接口，然后用 Java2WSDL 的 A-

pache Axis 工具生成 WSDL poteType 接口描述文件。但复杂的接口不能总是正确地转换为 WSDL。

2. 直接编写 WSDL portType 接口文件。这是最通用的选择。如果直接编写 WSDL，就可以完全控制服务 PortType 的描述。然而由于 WDSL 是一个相当繁琐的语言，因此它不是很友好的。

### 3.2 生成 stub 文件

网格编程是一种分布式的编程模式，客户端和服务器之间的通信和调用由 stub 负责。客户端对服务器端的网格服务的调用是通过本地 stub 进行，而服务产生的结果由服务器 stub 返回给调用方。GT3专门提供了产生 stub 的编程工具，例如：用 GT3 命令生成 stub 文件，java org. globus. ogsa. tools. wsdl. GSDL2Java *. wsdl。这样就自动生成了 stub 程序的 java 文件。

### 3.3 实现服务

网格服务的实现是满足一定要求的 Java 类。在这个类中将为服务接口方法(Math. java)提供声明，还能提供额外的不能通过网格服务使用的 private methods。实现服务与 Java 中的编程模式完全相同。

所有的网格服务都必须继承 GridServiceImpl 这个基类(这是通常的框架类，因为它包含了所有网格服务共有的基本的功能)。

### 3.4 创建部署文件

部署阶段的一个关键部分是部署描述器(the deployment descriptor)。这个文件告诉 Web 服务器如何发布我们的网格服务(例如，告诉 Web 服务器网格服务的 URL 是什么)。部署描述器被写成 WSDD(Web Service Deployment Descriptor)格式。部署文件包含网格服务完整的 GSH，指定可以公开访问的接口定义中的方法以及网格服务的实现是由哪个 Java 类提供等信息。

### 3.5 部署服务

通过上面的编程步骤后，就可以对网格服务进行编译。GT3专门提供了编译和打包的工具来得到最后的 gar 文件。要部署该服务，就必须在服务器上运行该 gar 文件，并启动服务器容器。最后编写客户端程序完成对该服务的访问。

**小结** 网格可视为一个网格功能服务的扩展集合。核心服务完成服务级别、数据访问和集成、工作流、安全、策略、监测和诊断等目标，基于核心服务之上可以构建高级服务。因此，针对不同类型的网格(计算、数据和服务等类型)，如何利用核心服务来构建特定应用的高级服务就是网格开发的关键之所在。OGSA 是当前网格服务概念事实上的通用标准，它只对网格服务应该完成的功能作出了概念上的规范，但对具体的技术实现却没有任何的约定。而 Open Grid Services Infrastructure(OGSI)就是对 OGSA 概念正式的技术性实现[3]。

## 参 考 文 献


1 Foster I, Kesselman C, eds. The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann Publishers, 1998
2 Foster I, et al. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration, 2002. 6
3 Tuecke S, Foster C I. Open Grid Services Ifrastructure (OGSI). http://www.ggf.org/ogsi-wg, 2003. 6
4 Ferreira L, Berstis V. Introduction to Grid Computing with Globus. ibm. com/redbooks, 2003. 09
5 Cano J. Step-By-Step Example for the Globus Toolkit 3. 0. http://usuarios.lycos.es, 2003. 02


(上接第72页)

信件头中的单词能够在计算中起到作用，这样很容易被判为垃圾邮件。在改进的贝叶斯算法中，应将中文和英文的单词库完全区分开来，并对中英文邮件采用各自的单词库来计算概率。

第三是建立单词库中的用户组。贝叶斯算法具有个性化的特征，很多基于该算法的工具都是安装在客户端。为了减少用户的操作和方便管理，本文的综合算法是应用在服务器端，但在使用贝叶斯算法时，必须保持它的个性化，主要体现在单词库上。根据实验环境的用户分类特征，可以将用户分为多个组，按组建立单词库。极端情况下，可以单个用户分为一个组。

本文中贝叶斯算法的检测率没有 Paul Graham[3]的工具的效率高，因为后者使用于英文环境，而本文所针对的是中文环境，并且没有使用手工的词库。如果能够对贝叶斯算法进行以上三方面的改进，尤其是中文单词的自动划分方法，系统整体的检测准确率可以进一步提高。

综合算法的优越性还体现在系统使用和管理的方便程度上。无论是黑白名单算法，还是基于规则的算法都需要用户不断地更新和维护操作，而单一的贝叶斯算法也必须依赖于用户的手动反馈，因此任何一种单一的算法都需要增加用户大量的额外操作。而综合算法可以进行自动的学习和工作，不需要用户甚至管理员过多的干预。用户的反馈不再是必须的，而只是受到鼓励的行为，当用户愿意反馈一些信息时，系统会在效率上给用户予回报。

**结束语** 本文针对目前日益严重的垃圾邮件问题，提出了一个适合中文邮件环境的垃圾邮件综合过滤方法。它以动态维护的白名单来保护正常邮件的收取，以规则算法作为辅助过滤手段，采用用户发出的邮件作为重要的学习资料来源，并将贝叶斯算法作为核心来提高邮件过滤的准确度，从而使整个系统可以在保护正常邮件的基础上，自动地帮助用户过滤垃圾邮件。

该综合过滤方法是根据真实的中文邮件环境特征，在对原始邮件数据进行实验分析和数据统计的基础上提出的，并采用实际数据进行了效率分析，具备较高的可信度和实用性。但该综合方法在实际应用中还需完成贝叶斯算法本身的改进，从而使系统的准确率更高。

本文的综合过滤方法仅针对内容上不被请求的垃圾邮件而言，对于蠕虫病毒引起的垃圾邮件不能很好地过滤。蠕虫邮件以传播自带的病毒附件为目的，很少带有大量文本内容，其发送方地址多为伪造地址，并且具有很强的周期性，因此对蠕虫邮件和内容上的垃圾邮件的过滤方法是完全不同的。

## 参 考 文 献


1 Graham P. A Plan for Spam. Aug. 2002. http://www.paulgraham.com/spam.html
2 Graham P. Stopping Spam. Aug. 2003. http://www.paulgraham.com/stopspam.html
3 Graham P. Better Bayesian Filter. Jan. 2003. http://www.paulgraham.com/better.html
4 Pantel P, Lin D. SpamCop-- A Spam Classification & Organization Program. In: Proc. of AAAI-98 Workshop on Learning for Text Categorization
5 Sahami M, Dumais S, Heckerman D, Horvitz E. A Bayesian Approach to Filtering Junk E-Mail. In: Proc. of AAAI-98 Workshop on Learning for Text Categorization