基于S-OGSA的语义网格服务资源的设计

魏化永,周鸣争
(安徽工程科技学院 计算机科学与工程系,安徽 芜湖 241000)

摘 要:在当前语义网格应用中,设计语义网格组件和应用需要有一个标准的参考模型或系统性的框架。讨论了基于语义开放网格服务框架(S-OGSA)的语义网格服务资源的设计。介绍了OntoGrid项目组提出的S-OGSA语义网格参考模型;论述了S-OGSA模型的体系结构及核心服务:语义产生服务、语义绑定服务、语义感知的网格服务。结合具体步骤,给出了利用Globus Toolkits4平台设计具有良好可伸缩性的语义绑定服务资源的过程,为开发语义网格应用和语义组件提供借鉴和参考。

关键词:语义网格;S-OGSA;元数据;语义绑定服务;网格服务
中图分类号:TP393 **文献标识码**:A **文章编号**:1673-629X(2009)02-0116-04

Design of S-OGSA-Based Semantic Grid Service Resource

WEI Hua-yong, ZHOU Ming-zheng
(Dept. of Computer Science & Technology, Anhui University of Technology and Science, Wuhu 241000, China)

Abstract: At present, it needs a reference architecture or any kind of systematic framework for designing semantic grid components or applications. Focuses on the design of S-OGSA-based gird service resource. It introduces a reference architecture of semantic grid, namely semantic-open grid service architecture(S-OGSA), which proposed by OntoGrid project. The points relate S-OGSA's model, the mechanisms and the core of its capabilities: semantic provisioning service, semantic binding service and semantic awared grid service. Through the concrete step, it provides the process that constructs the semantic binding service-resource of the good flexible function based on the Globus Toolkits 4 platform. This approach provides a reference to others in developing semantic grid application and semantic components.

Key words: semantic grid; S-OGSA; metadata; semantic binding service; grid service

0 引 言

网格已经从最初的计算网格,通过与语义网和语义Web Service技术的融合发展成为面向服务的智能语义网格[1]。如何从网格到语义网格的移植是一个重要研究课题,并得到了广泛研究。目前比较有影响的有OntoGrid的S-OGSA参考体系结构[2],myGrid的基于Web Service的一套工具集[3],InteliGrid项目的基于GlobusToolkit的InteliGird语义网格体系结构[4]。它们为语义网格应用和语义网格组成部件的设计提供了参考方法。在研究S-OGSA的基础上给出了语义网格服务资源的开发设计。

收稿日期:2008-05-22
基金项目:国家自然科学基金(60574028);安徽省自然科学基金(070412058);安徽高校省级自然科学研究重点项目(KJ2007A046)
作者简介:魏化永(1976-),男,安徽宿州人,硕士研究生,研究方向为网格计算、语义网格、Web服务;周鸣争,教授,研究方向为计算机网络与信息安全。

1 S-OGSA体系结构

S-OGSA是OntoGrid项目提出的一个语义网格参考体系结构,它是OGSA的扩展,如图1所示。它在网格中引入了语义的概念,是网格服务和语义网格服务混合的生态系统,为构建语义网格应用提供中间件服务。

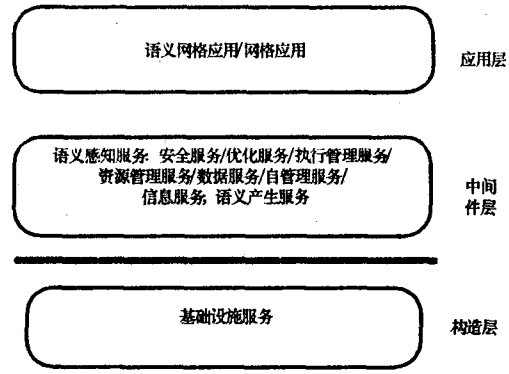

图1 S-OGSA在OGSA中的表示

S-OGSA主要讨论了:信息模型、应用机制和服

务能力。信息模型中包含一些新的实体及各实体间的关系;服务能力,模型中各实体所能提供的新服务;应用机制:描述如何把该体系结构应用于具体的网格应用平台,其中的语义服务实体如何表示等。

1.1 S-OGSA模型中的实体和服务的关系

在网格环境下,用明确的语义来表达知识,而这些知识的来源可以渗透于如图1所示的所有层次。在S-OGSA体系结构中:中间件层和构造层的语义表示是研究的重点。S-OGSA模型引入了网格实体、知识实体和语义绑定等新的概念,它们三者之间的关系[6]如图2所示。

1.1.1 S-OGSA中的相关概念

Grid Entities:网格实体,网格中一切都是网格实体,它具有一个唯一身份标识,包括网格服务和网格资源。

Knowledge Entities:知识实体,是一种特殊类型的网格实体,是可以用各种形式表示和操作的知识。

Semantic Bindings:语义绑定,是表述网格实体和知识实体之间关系的一种实体。它提供网格实体到语义网格实体的映射;在模型中,语义绑定首要的是它被视为一个网格资源,它具有可确定性和可管理性。如图2所示:语义绑定是知识实体和语义网格实体的桥梁。

Semantic Grid Entities:语义网格实体,是具有语义的网格实体。

1.1.2 S-OGSA的服务及其之间的关系

S-OGSA模型主要由如下2个核心服务组成:模型中的实体和服务之间的关系[5],如图2所示。

(1) Semantic-Provisioning Services:语义产生服务,通过注释现存的网格服务或资源动态地创建元数据。它提供对不同格式的知识实体和语义绑定进行管理和访问服务。主要分为两类服务:

a. 知识生成服务:负责管理描述知识的概念模型(如存储、访问和推理服务),主要包括本体服务和推理服务等。

b. 语义绑定生成服务:主要包括元数据服务和注释服务。

(2)Semantic Aware Grid Service(SAGS):语义感知的网格服务,是指在网格环境中利用知识技术来增强其服务能力的各类网格服务。所谓语义感知是指,在基于知识和元数据的基础上,能够理解绑定服务并采取相应的操作(如:在语义资源目录中,对所有实体进行搜索请求)。如图1中所示的服务都是SAGS。

1.2 S-OGSA的应用机制

把上文所述的一些概念,应用于具体的网格平台,要求实现须具有网格实现系统平台独立性,满足语义OGSA所定义的"概念性、节约性、一致性"等规范。为此,S-OGSA定义了如下机制:

(1) 在模型中所有网格实体和语义绑定都视为网格资源,这使得各种各样的资源描述具有一致性。

(2) 有状态的语义服务:通过网格实体传输语义绑定,虚拟的网格资源具有明确的语义信息。可用WSRF[6]技术来进行描述和实现该服务。

2 基于WSRF实现的语义绑定服务资源

有状态的语义服务的实现是基于S-OGSA的语义网格应用开发的关键。下面以一个语义绑定服务资源SBS(Semantic Binding Service)为例,描述在GT4平台下开发一个语义服务资源的基本开发模式。SBS服务的主要功能是:通过客户端发送请求由SB(SemanticBinding)工厂服务生成一个本体SB,并进行存储,然后客户端可以对该本体进行查询服务。SBS主要括:Client, Semantic Binding Factory Service, Semantic Binding Service, Metadata Query Service, SB-Resource这五个部分。它们之间的关系如图3示。

2.1 程序设计

SBS服务系统实现的运行环境采用WindowsXp+Java+GT4平台,用GT4平台的Gloubs Java WS Core作为Web服务器,它提供了WSRF的实现;整个系统的编译采用了Aache Aant。

(1) 环境变量设置:

①JAVA_HOME=d:/jdk1.4.2(JDK的安装路径)

②ANT_HOME=d:/ant1.7.0(ANT的安装路径)

③GLOBUS_LOCATION=d:/gwsc(GT4的安装路径)

④Path=%JAVA_HOME%\bin;%ANT_HOME%\bin;%GLOBUS_LOCATION%\bin;%Path%(命令搜索路径)

(2) 服务接口:系统提供的三个服务的接口定义如下:

* 语义绑定工厂服务中定义生成一个SB实例:

```
public interface SBFactory {
SemanticBinding create(String contentRDF, String SB_R_id)
throws Exception; //创建一个SB;contentRDF是用WSRF服务
生成的RDF表示的SB的内容体,SB_R_id是SB的身份标识
}
```

* 在SemanticBinding服务中只定义了执行查询服务一个基本操作:

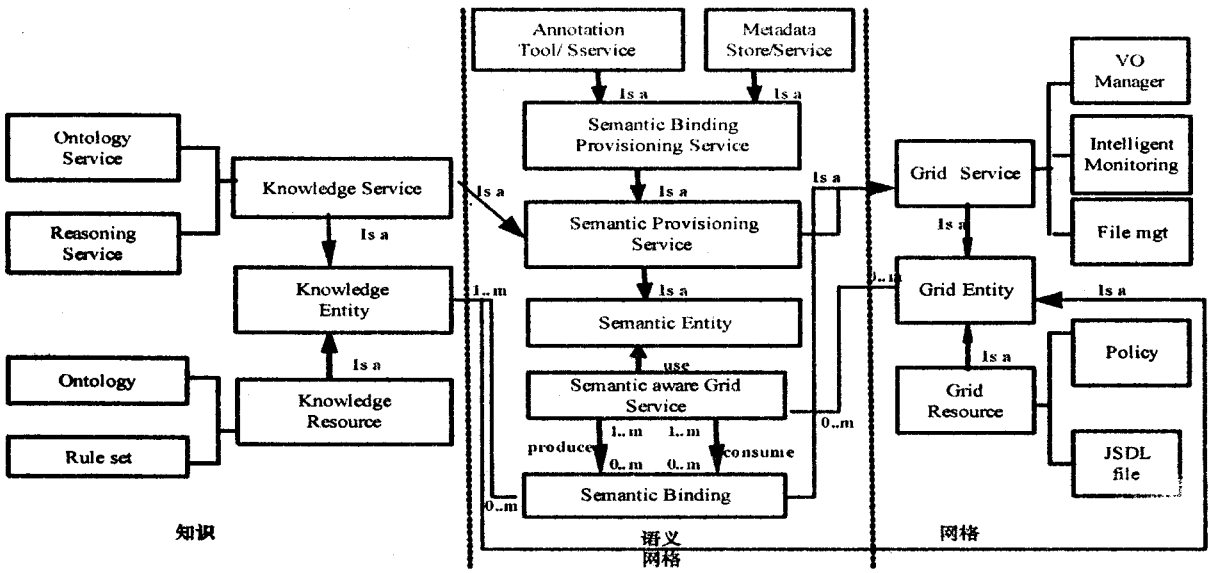

图 2 语义网格信息模型的扩展视图

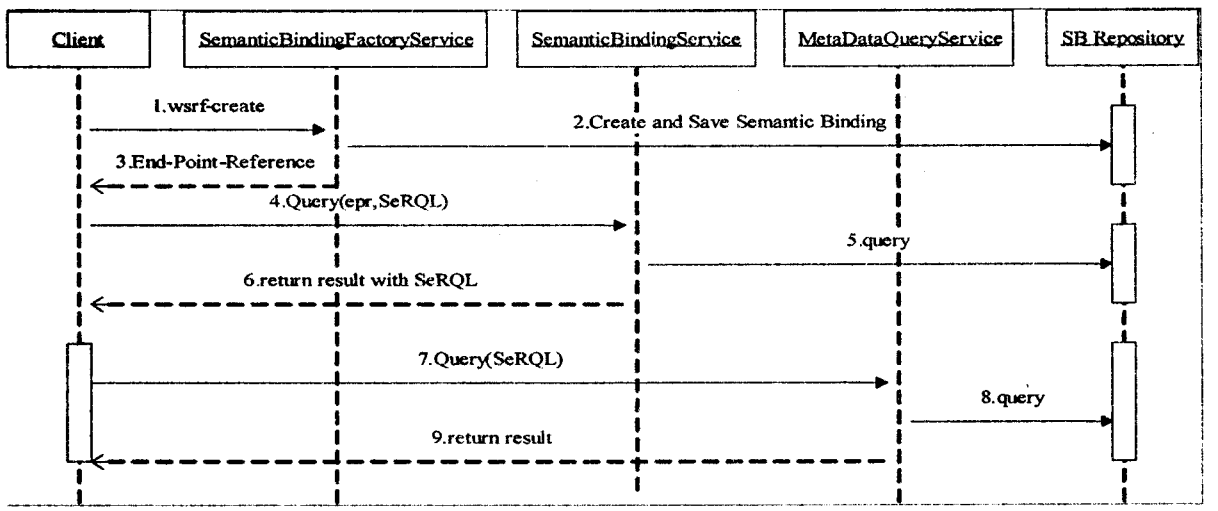

图 3 语义绑定服务系统序列图

```
public interface SemanticBinding {
    public String miniQuery(String query);//对 SB 执行查询 query,返回查询结果
    }
```

* 在 MetadataQueryService 服务中接口定义如下:

```
public interface MetadataQueryService extends javax.xml.rpc.Service{//返回查询端口类型及地址
    Public String
    getMetadataQueryPortTypeAddress();
    //返回元数据查询端口类型
    public MetadataQueryPortType
    getMetadataQueryPortType() throws
    javax.xml.rpc.ServiceException;
    //返回元数据查询端口类型
    public MetadataQueryPortType
    getMetadataQueryPortType(java.net.URL portAddress)
    throws
    javax.xml.rpc.ServiceException;
    }
```

(3)服务及资源的实现:在得到能够准确描述服务的接口文件后,还只能说服务能够做什么了,但还没有指定它怎么做。这里,只给定 Semantic Binding 服务的实现服务和资源的详细步骤如下:

1) 定义 QNames 接口:当用户必须引用与这个服务相关的所有事物时,需要使用 qualifiednames (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 ResourceHome 类:实现有

状态资源的 ResourceHome 可从 GT4 包括的 SingletonResourceHome 类继承实现。这个基类几乎提供了所有 ResourceHome 类用来管理单个资源的功能,只需要实现 findSingleton 方法。当用户第 1 次请求资源时,这个方法将会被 ResourceContext 类调用。findSingleton 方法可建立一个新的资源对象、初始化它、返回它。SingletonResource 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: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 服务接口提供声明。

```
public class SemanticBindingServiceImpl implements SemanticBindingPortType{
    //查询操作
    public String miniQuery(java.lang.String inParam) throws java.rmi.RemoteException{
    SemanticBindingResource sb_resource = getResource();
    String res=sb_resource.miniQuery(inParam);
    return res;}
    //通过 getResource()方法获得对一个资源的引用
    private SemanticBindingResource getResource() throws RemoteException{//对资源上的有状态信息进行各种操作
    Object resource = null;
    try{resource=ResourceContext.getRescurceContext().getResource();
    }catch(Exception e){throw new RemoteException("访问资源上下文时异常",e);}
    SemanticBindingResource sb_resource = (SemanticBindingResource)resource;
    return sb_resource;}}
```

2.2 部署服务与资源

至此,SBS 系统服务和资源所需要的服务接口、Qnames 文件、WSDL 文件、Resource 文件、ResourceHome 文件和服务实现的全部 Java 代码已经编写完毕,接着需要编译所有的代码并把它们部署到 Web 容器。

部署 SBS 系统的具体步骤如下:

(1)编译 Java 程序代码:在 MS-DOS 命令窗口中执行命令 ant dist 编译整个工程,会创建生成 Globus Archive(GAR)文件包。

(2)部署 GAR 包:在上一步创建 GAR 文件后,就可继续执行命令 ant deploy,把 GAR 文件部署到 Globus WSRF 服务容器。

(3)取消部署 GAR 包:ant undeploy 命令将从 Globus 容器中删除工程相关的文件。

2.3 功能测试

S1. 启动服务器端容器程序:Globus-start-container-nosec

S2. 系统功能进行测试:首先要生成一个 SB,然后对 SB 的进行查询,读取 SB 属性并输出。

①antcreate_sb-Dparam=http://localhost:8080/wsrf/services/SemanticBindingFactoryService 命令执行后,将生成一个 SB 资源,End-Point-Reference 信息写进在当前工作目录创建的 SB_ept.txt 文件中。

②ant query_sb -Dparam=.\SB_epr.txt,从 SB_epr.txt 文件中进行查询,并在终端输出查询结果

③ant read_props -Dparam=.\SB_epr.txt,读取 SB_epr.txt 文件,并在终端输出 SB 的 epr 信息。

3 结束语

S-OGSA 是扩展的 OGSA,其中所有语义实体都是遵循 OGSA 的开放式服务,使程序员可以方便地声明和实现语义网格服务,从而设计出更灵活、更商业化的语义网格应用程序,这是对传统语义网格开发模式提出的挑战。理解 S-OGSA 体系结构模型是语义网格开发的基础,遵循 OGSA 的服务开发的基本模式,是语义网格服务程序设计的逻辑指导,在语义网格应用研究中,S-OGSA 为建造从网格应用到语义网格应用提供了较好的参考。

基于该框架开发校园网格应用是下一步研究的方向。

参考文献:

[1] De Roure N R S, Jennings D. The Semantic Grid: Past, Present and Future[J]. Proceedings of the IEEE,2005,93(3):669-681.

[2] Corcho O, Alper P. S-OGSA. Semantic Reference Architecture[EB/OL]. 2006-06. http://www.ontogrid.net/ontogrid/r_semantic.jsp.

[3] Turi D. Deployment Architecture in the myGrid[EB/OL]. 2006-03. http://www.mygrid.org.uk/wiki/Mygrid/DeploymentArchitecture.

[4] Inteligrid Semantic Aarchitecture[EB/OL]. 2007-08-01. http://www.inteligrid.com/inteligrid-semantic-grid-architecture.htm.

[5] Corcho O, Alper P, Kotsiopoulos I, et al. An overview of s-ogsa: a reference semantic grid architecture[J]. Journal of Web Semantics, 2006,4(2):102-115.

[6] Czajkowski K, Ferguson D. Web Services Resource Framework(WSRF)[R].[s.l.]:Globus Alliance and IBM,2005.