

密级：

论文编号：

中国农业科学院 学位论文

基于 OGSA 农业科技数据库集成研究

**Research on OGSA-Based Agricultural Sci-Tech Database
Integration System**

硕士研究生：郭曼

指导教师：王文生 研究员

申请学位类别：管理学硕士

专 业：管理科学与工程

研 究 方 向：信息管理与信息系统

培 养 单 位：农业信息研究所

研究生院

提交日期 2007 年 06 月

Secrecy:

No.

Chinese Academy of Agricultural Sciences

Master Dissertation

Research on OGSA-Based Agricultural Sci-Tech Database Integration System

Ms. Candidate:	Guo Man
Advisor:	Professor Wang Wensheng
Degree:	Master of Management Science
Major:	Management Science and Engineering
Specialty:	Management Information System

Chinese Academy of Agricultural Sciences

June 2007

中 国 农 业 科 学 院

硕士学位论文评阅人、答辩委员会名单表

论文题目		基于 OGSA 农业科技数据库集成研究与实现				
论文作者		郭曼	专 业	管理科学与工程	研究方向	信息管理与信息系统
指导教师		王文生 研究员		培养单位（研究所、中心）		农业信息研究所
姓名		职称	硕（博） 导师	单 位	专 业	签 名
评 阅 人	曹存根	研究员	硕导□ 博导√	中科院计算技术研究所	人工智能、知识工程、 大规模知识处理	
	诸叶平	研究员	硕导√ 博导□	中国农业科学院农业信息 研究所	农业信息技术研究	
答 辩 主 席		沈佐锐	教授	硕导□ 博导√	中国农业大学	生物物理
答 辩 委 员	卢兵友	研究员	硕导√ 博导□	科技部农村技术中心	农业信息技术	
	曹存根	研究员	硕导□ 博导√	中科院计算技术研究所	人工智能、知识工程、 大规模知识处理	
	杨宝祝	研究员	硕导□ 博导√	国家农业信息化工程技术 研究中心	农业信息技术	
	刘继芳	研究员	硕导√ 博导□	中国农业科学院办公室	资源区划	
会议记录（秘书）			于迎建 副研究员			
论文答辩时间地点			2007 年 6 月 21 日，下午 2：00 在 420 会议室			

独 创 性 声 明

本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得中国农业科学院或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

研究生签名：

时间： 年 月 日

关于论文使用授权的声明

本人完全了解中国农业科学院有关保留、使用学位论文的规定，即：中国农业科学院有权保留送交论文的复印件和磁盘，允许论文被查阅和借阅，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。同意中国农业科学院可以用不同方式在不同媒体上发表、传播学位论文的全部或部分内容。

(保密的学位论文在解密后应遵守此协议)

论文作者签名：

时间： 年 月 日

导师签名：

时间： 年 月 日

摘 要

农业科技数据库集成研究的主要目标是研究符合农业科技数据特点的海量分布式异构数据资源的共享。在数据网格这样一个松耦合的分布式异构系统中，在不破坏数据资源管理者自治性的条件下，如何对这些共享的海量异构数据资源进行有效的查询是一个极具挑战性的课题。面对如此挑战，本文基于 OGSA 的农业科技数据的实际情况对分布式集成系统进行初步研究。

全篇论文以分析农业科技数据现存的问题开篇，分析农业科技数据的自身特点，简要介绍目前关于数据集成的普遍方法，并且从实际出发，估计农业科技数据在具体集成实施中面临的挑战。

在基础理论研究方面，文章着重从 Globus Toolkit 和 OGSA（开放式网格服务体系结构，Open Grid Service Architecture）等技术方面入手，分析开源网格基础平台 Globus 的更新、发展，并从中选择适用于本篇论文的基础平台；在介绍 OGSA、OGSA-DAI（开放式网格服务体系结构-数据访问集成，Open Grid Service Architecture – Data Access Integration）以及 Web Services 方面，作者从以往阅读过的文献资料以及书籍中消化、提取与本文相关的理论背景，为下文设计做好铺垫。

在农业科技数据库集成设计方面，作者根据数据库集成可能面临的问题提出解决方案，提出作业流程设计、编程设计、功能设计、服务交互设计以及应用架构设计。该设计涉及数据库集成问题的几乎各个环节，以期完成预想结论。

在选取理论与实践结合点的时候，文章选择农产品价格数据集成领域。运用数据集成理论将 FAO 叙词表与中华人民共和国农业部“菜篮子”工程结合。此结合将有效的利用国内外农产品价格数据，并从该类数据中提取与价格相关的各类信息。

关键词：农业科技数据，开放式网格服务体系结构，数据网格，数据库集成，农产品价格

Abstract

The major research objective of Agricultural Sci-tech Database Integration is to share the mass heterogeneous data in a distributed model. In a coupled grid-based distributed system, it becomes a challenging problem to query mass distributed and heterogeneous data effectively without exceeding the autonomy of data resource authority. To solve this problem, this thesis puts its emphasis mainly on the research of the distributed query processing technologies in a practical Data Grid environment.

On the analysis of existing problems of Agricultural Sci-tech Database, the thesis begins with characteristics of the agricultural data, introduces the current data integration methods briefly, and estimates the challenges of integration of agricultural sci-tech data in reality.

In the basic theory research, the thesis investigates the Globus Toolkit, OGSA (Open Grid Service Architecture) and other technology aspects. After analysis of the 3 generation evolution of Globus Toolkit, the thesis chooses the most suitable one as its computing platform. In order to build firm theoretical foundations, the introductions of OGSA, OGSA-DAI (Open Grid Service Architecture – Data Access Integration) and Web Services come from digestion and extraction of the previous readings.

According to the potential problems facing database integration, on the design of Agricultural Sci-tech Database Integration, this thesis proposes a solution scheme, and designs the process, programming, foundations, services interaction and application architecture. To achieve the desired conclusion, these designs are related to almost every aspect of database integration.

The thesis takes an effort to choose a research instance, in which we can make a good use of Agricultural Sci-tech Database Integration. It chooses Agricultural Products Price Database Integration as a research instance, and combines with the databases of AGROVOC (FAO) with Marketing Project (MOA). This combination will make full use of the Agricultural Product Price Data and extract the price-related information, which proves that this research can be effective in the domestic and foreign agricultural marketing areas.

Key word: Agricultural Sci-Tech Data, OGSA, Database Integration, Agricultural Product Price

目 录

第一章 绪论.....	1
1.1 研究背景.....	1
1.1.1 国外数据网格研究	1
1.1.2 国内数据网格发展	2
1.2 研究问题的提出	3
1.3 研究方法.....	3
1.4 研究面临的问题	4
1.5 研究目标.....	5
1.6 研究内容.....	5
第二章 基于 OGSA 的数据库集成技术研究	6
2.1 开源网格基础平台 Globus 简介	6
2.1.1 Globus Toolkit 2.....	6
2.1.2 基于 OGSi 的 Globus Toolkit 3	6
2.1.3 基于 WSRF 的 Globus Toolkit 4.....	7
2.1.4 WSRF 与 OGSi 比较	7
2.2 开放式网格体系结构	9
2.2.1 OGSA 基本思想.....	9
2.2.2 OGSA 的主要架构分析	10
2.2.3 OGSA 端口类型.....	11
2.2.4 OGSA 的服务接口	12
2.3 OGSA-DAI	13
2.3.1 OGSA-DAI 基本思想.....	13
2.3.2 OGSA-DAI 的数据服务结构	14
2.3.3 OGSA-DAI 端口类型.....	15
2.3.4 OGSA-DAI 支持的数据库	16
2.4 Web Services.....	17
2.4.1 Web 服务的定义及特征	17
2.4.2 Web 服务的工作原理	18
第三章 基于 OGSA 农业科技数据库集成 (AST-DB) 设计.....	19
3.1 AST-DB 集成的作业流程设计	19
3.2 AST-DB 集成的编程设计	20
3.2.1 AST-DB 集成中的服务器端组件	22
3.2.2 AST-DB 集成中的客户端组件	22
3.2.3 AST-DB 编程设计小结	23
3.3 AST-DB 集成的功能设计	23

3.3.1 AST-DB 生命期管理功能	23
3.3.2 AST-DB 服务注册 / 取消注册功能	23
3.3.3 AST-DB 服务发现功能	23
3.3.4 AST-DB 服务通知功能	24
3.4 AST-DB 的数据服务交互	24
3.5 AST-DB 集成的应用架构设计	26
3.5.1 AST-DB 数据采集层	26
3.5.2 AST-DB 数据处理层	26
3.5.3 AST-DB 数据应用层	27
第四章 AST-DB 农产品价格数据集成系统研究与设计	28
4.1 农产品价格数据 AST-DB 集成概要设计	28
4.1.1 目的	29
4.1.2 应用需求分析	29
4.1.3 主要业务流程	30
4.1.4 主要业务用例	31
4.1.5 数据服务设计	32
4.2 农产品价格数据集成 AST-DB 访问管理	35
4.2.1 部分网格数据管理权限的用户访问管理	35
4.2.2 全局网格数据管理权限的用户访问管理	35
4.3 农产品分类与多语言转换 AST-DB 网格服务数据访问	35
4.3.1 单数据库网格数据服务访问	36
4.3.2 双数据库网格数据服务访问	37
4.3.3 多数据库网格数据服务访问	38
4.4 农产品价格 AST-DB 网格服务意义	39
第五章 总结与展望	41
5.1 工作总结	41
5.2 下一步工作	41
附录	42
参考文献	45
致谢	50
作者简介	51

图 目 录

图 1	OGSI 与 WSRF 位置对照	7
图 2	在 WSRF 中 Web 服务与有状态资源相结合	8
图 3	OGSI 中有状态网格服务与服务数据相结合	9
图 4	开放式网格服务体系结构图	10
图 5	OGSA 网格服务结构	12
图 6	OGSA-DAI 在 OGSA 中的位置	13
图 7	OGSA-DAI 结构的总体示意图	14
图 8	访问 OGSA-DAI 中的数据	15
图 9	Web 服务工作原理	18
图 10	基于 OGSA 的 AST-DB 集成作业流程	19
图 11	实现基于 OGSA 的 AST-DB 集成应用中的数据流控制	21
图 12	基于 OGSA 的 AST-DB 的服务端组件	22
图 13	基于 OGSA 的 AST-DB 的客户端组件	23
图 14	AST-DB 中的服务通知	24
图 15	基于 OGSA 的农业科技数据库集成服务之间交互	25
图 16	基于 OGSA 的农业科技数据库 (AST-DB) 集成的应用架构示意图	26
图 17	2004-2006 年农产品进出口额统计	28
图 18	AGROVOC 截图	29
图 19	全国主要菜篮子产品批发价格日报价格列表	30
图 20	FOA_AGROVOC、MOA_Marketing 以及山东寿光蔬菜基地数据库集成示意图	31
图 21	农产品价格信息实体关系图	33
图 22	实体与 XML 架构对应关系图	33
图 23	农产品价格概要访问	36
图 24	单数据库网格数据服务访问	37
图 25	双数据库网格数据服务访问	37
图 26	多数据库访问—以汉语作为检索词语种	38
图 27	多数据库访问—以英语作为检索词语种	39
图 28	多数据库访问—以法语作为检索词语种	39
图 29	农产品价格关联因素示意图	40

表 目 录

表 1	OGSI 和 WSRF 功能对照	9
表 2	网格服务接口说明	12
表 3	OGSA-DAI 支持数据库类型说明	16

英文缩略表

英文缩写	英文全称 / URL	中文名称
AST-DB	Agriculture Sci-Tech Database	农业科技数据库
Apache Ant	URL: http://ant.apache.org/	Ant 是基于 Java 的自动化软件构造过程的工具，它类似于 make，能自动编译文件间互相依赖的程序
Apache Axis	URL: http://ws.apache.org/axis/	Axis 是用来和 SOAP 交换报文的 SOAP Web 服务引擎
Apache Cocoon	URL: http://cocoon.apache.org/	Cocoon 是一种 Web 应用开发框架，它实现了“组建管道”的思想，每一个组件在管道中特化为一种特殊的操作
Apache Jakarta Tomcat	URL: http://jakarta.apache.org/tomcat/	Tomcat 是一个 Java Servlet 容器，用来管理 servlet 和 JSP 的运行环境
Apache Jakarta Turbine	URL: http://jakarta.apache.org/turbine	Turbine 是一种基于 Java servlet 的框架，允许开发人员创建安全的 Web 应用程序
Apache Jakarta Velocity	URL: http://jakarta.apache.org/velocity	Velocity 提供了一种相对于 JSP 的可选择的脚本语言来开发 Web 页面
Apache Jetspeed	URL: http://portals.apache.org/jetspeed-1/	Jetspeed 是一种使用 portlet 创建 Web 门户的框架
ASAP	Asynchronous Service Access Protocol	异步服务访问协议，ASAP 是一种在客户机和服务器之间进行异步通信的协议
BPML	Business Process Modeling Language	业务流程模型语言，BPML 是一种用于业务流程建模的元语言
CA	Certification Authority	权威证书，CA 是发布和管理安全证书以及公共密钥以保证安全通信的权威部门
CCA	Common Component Architecture URL: http://www.cca-forum.org/	通用组建体系结构，CCA 是一种用于高性能计算应用的组建模型
CIM	Common Information Model URL: http://www.dmtf.org/standards/cim	通用信息模型，CIM 提供了一种通用的定义，这样，在整个网络

		环境中系统间的各部件可交换管理信息
Condor	URL: http://www.cs.wisc.edu/condor/	Condor 是一种资源管理和作业调度系统
CORBA	Common Object Request Broker Architecture	公共对象请求代理体系结构, CORBA 是一种创建分布式客户/服务器应用的中间件技术, 其中客户和 CORBA 对象之间不受地域、平台以及编程语言的约束
DAISGR	DAIServiceGroupRegistry	数据访问集成服务注册
DAML	DARPA Agent Markup Language URL: http://www.daml.org/	DARPA 代理标记语言, 基于 RDF 之上, DAML 是一种基于 XML 的、开发语义网的语言
DCE RPC	Distributed Computing Environment RPC	分布式计算环境 RPC, DCE RPC 是一种来自开放软件基金会 (OSF) 的 RPC 实现
DCOM	Distributed Component Object Model	分布式组件对象模型, DCOM 是一种创建基于 Windows 的分布式客户/服务器应用的中间件技术, 其中客户和 DCOM 组件之间的通信不受地域和编程语言的限制
ECS	Element Construction Set	元素构造集, ECS 是一种 Java API, 用来产生各种各样标记语言的元素, 包括 HTML 4.0 和 XML 在内
EJB	Enterprise JavaBeans	EJB 是 J2EE 框架中服务器方的一种技术
GARA	Globus Architecture for Reservation and Allocation	Globus 预约和分配体系结构, GARA 是 Globus 工具箱的一个组成部分, 用来对资源进行预约和分配
GDS	Grid Data Service	网格数据服务
GGF	Globus Grid Forum URL: http://www.gridforum.org/	全球网格论坛, GGF 是网格技术的权威机构
GIS	Grid Information Service	网格信息服务, GIS 是 Globus 工具箱的一个组成部分, 用来管

		理资源信息
Globus Toolkit	URL: http://www.globus.org	Globus 工具箱提供了一些创建网格系统的中间件技术工具
GMA	Grid Monitoring Architecture	网格监控体系结构, GMA 是 GGF 对监控网络的报告建议书
GRAM	Globus Resource Allocation Manager	Globus 资源分配管理器, GRAM 是 Globus 工具箱的组成部分, 用于作业提交
Grid Portal		网格门户是一种通过基于 Web 的用户接口来访问网格资源的系统
GridSphere	URL: http://www.gridsphere.org	GridSphere 是一种用 portlet 创建的 Web 门户的框架
GSFL	Grid Services Flow Language	网格服务流语言, GSFL 是一种基于 WSFL 的工作流语言, 用于兼容 OGSA 的网格服务的合成
GSH	Grid Service Handler	网格服务句柄, GSH 是一个全球唯一的 URL, 用来标识网格服务或网格服务实例
GSI	Grid Security Infrastructure	网络安全基础结构, GSI 提供了在网格环境中安全通信的基础。它基于公钥加密、X.509 证书以及 SSL 通信协议
GSR	Grid Service Reference	网格服务参考, GSR 是与网格服务或网格服务实例的实现相关联的参考
GT2	Globus Toolkit 2.x	Globus 工具箱 2.x 版, GT2 是用 C 语言来实现的
GT3	Globus Toolkit 3.x	Globus 工具箱 3.x 版, GT3 建立在 OGSI 和 Web 服务之上
GT4	Globus Toolkit 3.9.x	Globus 工具箱 3.9.x 版, GT4 建立在 WSRF 之上
GWSDL	Grid Web Services Description Language	GWSDL 是 WSDL 的扩展, 用来描述网格服务
IETF	Internet Engineering Task Force Internet URL: http://www.ietf.org/	工程任务组, IETF 是 Internet 技术的权威机构
J2EE	Java 2 Platform Enterprise Edition URL: http://java.sun.com/j2ee/index.jsp	Java 2 平台企业版, J2EE 定义了采用 Java 创建企业应用的标准

Java CoG Kit	Java Commodity Grid Kit	Java CoG 工具箱，Java CoG 是一个使用 Java 访问 Globus 资源的软件工具箱
JRMP	Java Remote Method Protocol Java	远程方法协议，JRMP 是 RMI 客户和对象使用的通信协议
JSP	Java Server Page	JSP 是服务器端脚本语言，其用来动态生成 Web 页面
MDS	Monitoring and Directory Service	监控和目录服务，MDS 是由 Globus 工具箱提供的一种信息服务
MDS2	Monitoring and Directory Service 2	MDS2 是由 GT2 提供的、基于 LDAP 协议的信息服务
MDS3	Monitoring and Directory Service 3	MDS3 是由 GT3 提供的、基于 OGSi 的信息服务
MJS	Management Job Service	作业管理器
MVC	Model-View-Controller	模型视图控制器，MVC 是一种用户接口管理系统模型，其将表现逻辑和应用逻辑分离开来
OASIS	Organization for the Advancement of Structured Information Standards URL: http://www.oasis-open.org	结构化信息标准促进组织，OASIS 是电子商务技术的权威机构
OGSA	Open Grid Service Architecture URL: http://www.globus.org/ogsa/	开放网格服务体系结构，基于 Web 服务之上，OGSA 是创建面向服务的网格系统的事实上的标准
OGSA-DAI	Open Grid Services Architecture-Data Access and Integration URL: http://www.ogsadai.org.uk/	开放网格服务体系结构-数据访问与集成，OGSA-DAI 是一种访问和集成源自不同数据源的数据（例如关系型和 XML 数据库系统）以及网格上的文件系统的中间件技术
OGSI	Open Grid Services Infrastructure	开放网格服务基础结构，OGSI 是一个实现 OGSI 定义的接口的规范说明。OGSI 已经由 WSRF 所替代

RDF	Resource Description Framework	资源描述框架，RDF 是一种基于 XML 的语言，用来描述结构化的元数据
RDFS	RDF Schema	资源描述框架计划，资源描述框架计划是对 RDF 的扩展，其具有更多的原型
RFT	Remote Method Invocation	远程方法调用，RMI 是创建基于 Java 的分布式客户/服务器应用的中间件技术，其中客户和对象的实现不受地域和平台的约束
RSL	Resource Specification Language	资源规范语言，RSL 是 Globus 工具箱使用的作业提交描述语言
SGE	Sun Grid Engine URL: http://gridengine.sunsource.net/	Sun 网格引擎，SGE 是一种资源管理和作业调度系统
SOA	Service Oriented Architecture	面向服务的体系结构，SOA 是开发松散耦合的、分布式系统的模型，其中软件组件不受服务的影响，它们可在网络环境中发布并且由客户发现
UDDI	Universal Description, Discovery, and Integration URL: http://www.uddi.org/	统一描述，发现和集成，UDDI 是在 Web 服务中服务注册和发现的工业标准
URI	Uniform Resource Identifier	统一资源标识符，URI 是对 Web 上的资源标识的标准方法
VO	Virtual Organization	虚拟组织，VO 是一种动态环境，它耦合了地理上分布的资源（它们也许运行在多个研究机构）。VO 也许拥有多条规则，来规定它的成员如何安全访问和共享它的资源
W3C	World Wide Web Consortium URL: http://www.w3.org/	万维网联盟，W3C 是 Web 技术的权威机构
WS	Web Services URL: http://www.w3.org/2002/ws/	Web 服务，WS 是基于 XML 的、创建面向服务的、分布式应用的中间件
WSDL	Web Services Description Language Web URL: http://www.w3.org/TR/wsdl	服务描述语言，WSDL 是基于 XML 的语言，其用来描述 Web

		服务接口
WS-Inspection	Web Services Inspection	Web 服务检查，Web 服务检查时 服务注册和发现的工业标准
WSRF	Web Services Resource Framework	Web 服务资源框架，WSRF 是模 型化提供 Web 服务的有状态的资 源的规范集
XML	Extensible Markup Language	可扩展标示语言，XML 是 Web 上表示结构化信息的一种标准文 本格式

第一章 绪论

农业是第一产业，是国民经济的基础。科学技术是第一生产力，是推动农业发展的主要动力。自十六届五中全会以来，“三农”问题、农业标准化、农业信息化等问题已成为农业发展的重要问题。如何使农业向产业化、市场化发展，向信息化、现代化靠近，是我国农业科研工作者长期探寻的课题。改革开放近三十年，我国农业领域发展突飞猛进，这与农业信息技术的发展密切相关。从建国初期，由少数农业科学家保存农业研究资料，到二十世纪六七十年代形成农业科技图书馆、农业信息中心，一直到改革开放近三十年共享基于信息技术的农业数据库，农业信息技术的出现为我国农业提供高效、安全、稳定的发展平台。

但是，随着农业发展经历改革开放的高速发展时期，目前农业领域也面临着许多发展中的瓶颈问题，尤其是数据资源问题。数据量庞大、数据冗余繁多、数据资源使用效率低等一系列亟待解决的问题。

1.1 研究背景

目前，数据网格在世界上许多国家和地区都得到了很大的重视和发展。随着数字化革命和 Internet 的大发展所带来的经济、贸易、信息传播的全球化，数据集容量呈爆炸式增长趋势。在一些大型科学研究、信息服务和数字多媒体技术等研究领域和应用领域中，数据集容量已经相当庞大并在继续增长。例如在全球气候模拟、海洋环流模拟、高能物理、核爆炸模拟、生物工程、国防信息建设、数字地球等应用中，它们的数据量将达到 TeraByte(1000GB)至 PetaByte(1000TB)的级别。数据网格的最终目标是建立异构分布环境下海量数据的一体化存储、管理、访问、传输与服务的架构和环境。它可以很好地解决海量数据难于组织、处理的问题。在推动应用发展的同时，数据网格本身也取得很大发展。

数据网格技术以数据管理为中心，面向底层屏蔽网络中各种异构存储的数据资源，面向上层应用提供通用和可靠的数据服务，可为地理上分布的研究团体对海量数据复杂分析、联合处理提供基本环境，单个研究人员可以充分调动网格上的计算资源、信息资源，方便地访问和分析庞大的数据。二十一世纪将是信息，特别是数字信息占主导地位的世纪，研究数据网格计算技术具有积极的学术意义和很高的应用价值。

1.1.1 国外数据网格研究

TeraGrid 项目是由美国国家科学基金（NSF）于 2001 年 8 月发起，当时投资 5300 万美元资助超级计算中心，构建计算网格。其目标是为开放的科学研究建立和部署一个世界上最大的、最全面的分布式计算基础设施。2003 年 9 月该项目宣称将增加更多的科学设备、大规模数据集、计算资源和存储资源，所有的成员之间将使用 40GB/S 的网络连接起来。该项目使用了 Globus Toolkit、Condor-G、MPICH-G2 等面向计算网格环境的软件，使得登录到该计算网格环境的用户可以透明地使用其中的计算、存储资源，进行大规模科学计算和数据处理。

IPG（Information Power Grid）项目是由美国 NASA（National Aeronautics and Space

Administration) 构建, 是为了实现网格计算实验床, 它可以将 NASA 分布在各地的计算和存储资源通过网络 (包括无线网络) 连接起来, 解决一些 NASA 目前无法解决的大规模的科学与工程计算和数据管理等问题。

PACI (Partnership for Advanced Computational Infrastructure) 项目由美国国家科学基金资助, 包括两个重要部分, 分别是 NCSA (National Computational Science Alliance) 和 NPACI (National Partnership for Advanced Computational Infrastructure)。目标是通过将学术界、政府部门和工业界的计算基础设施联合起来, 建立一个网格计算基础设施的伙伴联盟, 促进科学发现、知识传递和工程开发。

Access Grid 项目由美国 Argonne 国家实验室发起, 吸引了多所大学和研究机构加入, 汇集的资源包括计算资源、多媒体显示、表现和交互式环境, 提供网格中间件和可视化环境的访问接口, 特别适用于组对组的通信交互。

欧洲数据网格 EDG (European Data Grid) 是一个国际性大型研究和技术发展项目, 由欧洲原子核研究组织 CERN 领导, 另外还包括欧洲各国的五个主要合作伙伴和欧洲各国的 15 个相关机构。EDG 的主要目的是发展并测试一种科研“合作”的技术基础设施, 使得研究者能够突破地理局限共同研究, 允许分布在世界各地的工作者交互、共享数据和设备。EDG 主要针对 CERN 的高能物理应用, 解决海量数据的分解存储和处理问题, 同时将之扩展到其他应用。其方案是在 Globus 提供大部分网格基础服务之上, 着重面向高能物理、地球观测和生物工程应用的研究, 开发网格中间件、应用软件、评估软件及实验床。工作集中于数据管理的处理, 致力于解决世界范围内的科学团体对共享海量数据、大规模数据集计算与分析的需求。

英国政府同样非常重视网格技术的研究, 他们投资 2 亿英镑左右支持网格研究项目, 并确定了用网格计算技术构建 e-Science, 计划联合许多大学、国家级研究所和工业界共同完成多学科的大规模科学研究的信息基础设施和环境。

日本 Data Farm 网格项目主要用于 Petabyte 数据量的高能物理实验数据的分析和处理, 与欧洲数据网格相连。网格技术已成为日本信息技术领域的基础设施类项目。

1.1.2 国内数据网格发展

中国国家网格项目 (CNGrid)。“十五”期间, 国家高技术研究发展计划 (“863”计划) 资助该项目, 旨在建立面向企业、高等院校、科研机构、政府部门的国家高性能计算环境。主要任务包括: 第一, 建设中国国家网格实验床 (CNGrid); 第二, 建立具有 4 万亿次以上计算能力的网格主结点; 第三, 开发支持网格应用以及网格系统运行的维护软件; 第四, 开发具有代表性的网格生产应用。

教育部的中国教育科研网格项目 (ChinaGrid)。该项目是 “十五” 国家 “211 工程” 公共服务体系 “CERNET” (中国教育科研网) 高速地区网和重点学科信息服务体系建设项目中的重要内容, 目标是在 2005 年建立聚合计算能力超过 15 万亿次量级的教育科研网格, 形成世界上最大的超级网格之一。该项目的任务是依托 CERNET 建立聚合、共享资源的公共服务平台, 实现计算资源、存储资源、数据资源、信息资源、专家资源的多方位共享, 全面提高我国教育信息化基础设施服务水平和高等教育科研水平。

1.2 研究问题的提出

从对农业信息技术领域的总体概况分析中,我们能够得出目前农业数据资源主要面临以下三方面问题¹:

第一,农业信息网点多,信息人员多,但信息采集面和发布渠道少。

根据2007年1月23日,中国互联网络信息中心统计数据表明,农民上网用户人数仅仅占我国网民总数的0.4%,网民总人数为13700万人(95%置信度下的置信区间为[13362万人,14038万人])。可见,我国农民中的绝大部分无法使用计算机,也没有条件上网,常规的信息传播并非处处通畅,由于“数字鸿沟”的阻隔,造成电脑、网络、信息服务距离农民还差“最后一公里”。

第二,网站作为信息发布平台多,但是作为应用平台少。

近年来,随着信息技术的快速发展,各地都不同程度地加强了信息基础设施建设。但信息资源建设远不完善,“重建设轻服务”和“重硬件轻软件”的现象比较普遍。信息资源匮乏不仅制约了信息服务的开展,而信息如果只是数据的话,就无法体现信息系统的投资回报。比如,有些地区的农业信息发布只有事后数据,缺乏预测数据和相应的决策支持能力,就无法为政府决策提供依据,适时向公众发布预测信息,引导农产品生产经营者及时采取措施规避市场风险。

第三,各涉农部门相对独立,缺乏有效的信息资源整合与共享。

经过多年的建设,农业部信息资源建设取得了很大成果,农业部本级和各级涉农部门相继建立了农业政策法规、农村宏观经济、农产品进出口、农产品价格、农产品供求、农业科技等数据库,初步构建了农业分析预测预警等数据仓库。据不完全统计,目前,已经建立了各类规模、标准的数据库一百多个。但与信息化进展情况相比较,仍存在明显的问题。一是资源开发建设明显滞后,突出表现在有价值的信息匮乏和信息结构不合理,缺乏指导性和权威性;二是资源开发建设标准不统一、不规范,已有资源共享性差;三是市场信息采集处理分析手段落后、力量薄弱,资源开发先天不足。

如何解决上述数据集成、信息共享建设过程中所面临的主要问题?又如何在解决这些现有问题的同时还能对未来农业信息化更长远的发展打下良好的基础?一个开放的、基于标准的面向服务应用架构提供了这样的基础,它不但可以将现有农业应用整合到统一的应用服务平台中,还可以为农业信息化的长久发展和社会主义新农村的建设提供可持续发展的基础架构。

1.3 研究方法

将原有分散的、异构的各职能部门的数据进行有效的集成是很多涉农机构面临的问题。当前,异构数据库集成一般有两种方法:

第一,将原有的数据移植到新的数据库系统中来,为了集成不同类型的数据,必须将一些非传统的数据类型转化成新的数据类型。许多关系数据库供应商提供了类似的功能。这种集成方式的缺点是随着数据库的升级,原来数据的相关应用软件,或者被废弃或者重新开发,以适应新的数据库系统。因此,通常移植到一个新系统不是一个实际的解决方案。

第二,利用中间件集成异构数据库,该方法并不需要改变原始数据的存储和管理方式。中间件层位于异构数据库系统(数据层)和应用程序(应用层)之间,向下协调各数据库系统,向上为访问集成数据提供统一数据模式和数据访问的通用接口。各数据库的应用仍然完成它们的任

务，中间件系统则主要集中为异构数据源提供一个高层次检索服务。

本篇论文着重采用第二种研究方法，重点讨论数据库如何集成到网格环境中，满足应用程序对数据的访问。(1) 利用基于 OGSA 的体系结构，结合网格化数据库的思想满足用户对数据库集成的需求；(2) 采用在用户和关系数据库之间加入一个仲裁 (Mediator)，方便用户使用，满足用户操作，提供性能，支持可插拔；(3) 利用 Web Services 的技术，实现对数据的透明访问，满足用户从不同站点获取数据的需求。

1.4 研究面临的问题

随着农业数据库资源建设的不断推进，各个结点的可用信息规模呈爆炸式增长，尽管分布式数据库系统的查询优化技术已经得到广泛的研究。但是由于数据网格是一个松耦合的分布式异构自治系统，因此在网格环境下，对数据资源的定位和查询处理面临着前所未有的问题，主要考虑如下：

第一，各涉农部门结点高度自治

各涉农部门结点具有各自的数据管理策略，各自完成本部门内部应用，同时对数据资源使用具有分布性和并行性。

第二，各农业数据库结点之间的连接的带宽不同，其传输速度可能会有很大的差异。另外网络环境不稳定，经常会出现结点之间连接不上以及连接中断的情况。

第三，各农业数据库结点资源异构

农业数据库集成是由数据网格结点和连接结点的网络组成的。各农业数据库结点资源异构性体现在两个方面：网络异构性和结点异构性。网络异构性是指结点之间可能采取不同的网络技术连接，但是，TCP/IP 协议族的出现使得采用不同底层技术的网络可以互联在一起，因此已经消除了网络技术的异构性，从 TCP/IP 的角度上看，连接各个结点之间的网络都是一个同构的虚拟网络。

各农业数据库结点资源异构性可以分成五个层次：类别异构性、体系结构异构性、配置异构性、软件系统异构性、用户需求异构性 (李效东，2002；李志刚，2005；周园春，2006)。

(1) 类别异构性

类别异构性是指结点是计算机、网络存储设备还是科学仪器。类别异构性导致了资源在性能特征上的高度差异。

(2) 体系结构异构性

体系结构异构性是指结点可能具有不同的体系结构，如 SMP、MPP 以及 Cluster 等，体系结构异构性导致相同的代码不能运行在不同结点上。

(3) 配置异构性

配置异构性是指各个结点可能具有不同的处理能力、内存能力以及磁盘空间，如两个 SMP 结点可能具有不同速度的 CPU，两个 MPP 结点可能具有不同的处理器数目等。

(4) 软件系统异构性

软件系统异构性是指结点可能具有不同的软件系统。在软件系统异构性之中，最重要的是操作系统的异构性，此类问题在农业部门涉及较少。

(5) 用户需求异构性

农业数据库结点可能具有不同需求，如两个结点可能对查询计划的执行有不同的要求。

1.5 研究目标

相对于农业科技人员普遍各自独立地从事科学研究而言，通过共享相关领域的专业技术、专门研究以及区域信息，能够协助农业科技人员从更大的范围和复杂性上对农业科技数据充分利用。农业科技数据服务集成的主要目的是利用数据网格技术，研究设计符合农业科技数据特点的海量分布式异构海量数据资源的共享。

农业数据库资源集成的研究目标如下：

第一，选择能够将农业信息化领域与农业服务领域相结合的研究点，以体现数据库集成价值；

第二，根据 Globus Toolkit 和 OGSA 体系结构，通过对其结构和功能研究，提出农业科技数据库（AST-DB）集成的各环节设计；

第三，在开放式网格服务体系结构中，如何“按需集成”，完成对多种不同类型数据库集成；

第四，在 Windows 2003 环境下尝试部署 Globus Toolkit 4.0 和 OGSA-DAI，并确定网格数据服务资源。

1.6 研究内容

本文首先介绍了数据网格的相关研究背景，分析农业科技数据现存的问题。通过分析、比较 Globus Toolkit 的发展，选择适合于本论文的基础平台，并研究设计农业科技数据库集成框架。最后，以农产品价格数据集成为例，说明农业科技数据库集成的基本方法与意义。

围绕本论文的研究目标，主要进行以下几方面的研究：

1、对本论文的研究背景进行分析，研究农业科技数据系统的概念与特征。阐述国内外与本论文相关的数据网格研究现状，并分析农业科技数据库集成存在的问题以及研究方法。

2、文章着重从 Globus Toolkit 和 OGSA 等技术方面入手，分析开源网格基础平台 Globus 的更新、发展，并从中选择适用于本篇论文的基础平台；同时介绍 OGSA、OGSA-DAI 以及 Web Services 等相关技术。

3、在农业科技数据库集成设计方面，作者根据数据库集成可能面临的问题提出解决方案，初步完成作业流程设计、编程设计、功能设计、服务交互设计以及应用架构设计。

4、文章最后以农产品价格数据集成为例，选择利用数据集成理论将 FAO 叙词表与中华人民共和国农业部“菜篮子”工程以及山东寿光蔬菜基地农产品价格数据库结合，用以说明农业科技数据库集成的基本方法与意义。

第二章 基于 OGSA 的数据库集成技术研究

2.1 开源网格基础平台 Globus 简介

OGSA 对什么是网格服务和它怎样满足下一代网格应用作了一个高层次架构描述, 而 OGSI 是对网格服务怎样工作给了一个详细的规范说明, WSRF 又进一步的重构和进化了 OGSI。而这些都只是网格体系结构的设计内容, Globus 承担了提供对这些设计的具体实现的任务。

Globus Toolkit 工具包来源于国际上最有影响力的与网格计算相关的项目之一 —— Globus 项目, 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。Globus Toolkit 是一个开放源码的网格基础平台, 基于开放结构、开放服务资源和软件库, 并支持网格和网格应用, 目的是为构建网格应用提供中间件服务和程序库。

Globus Toolkit 具有较为统一的国际标准, 有利于整合现有资源, 也易于维护和升级换代。现在, 一些重要的公司, 包括 IBM 和微软等都公开宣布支持 Globus Toolkit。目前大多数网格项目都是采用基于 Globus Toolkit 所提供的协议及服务建设的。随着技术的发展和进步, Globus 体系结构也经历了几次飞跃, 现在它已经变得越来越完善。

2.1.1 Globus Toolkit 2

自从 1997 年起, Globus Toolkit 工具包的第二版 (GT2) 成为网格计算的事实标准。它强调可用性和互操作能力, 定义和实现了一些协议、API 和服务。

当时在世界上有为数众多的网格应用基于 GT2 平台之上, 通过提供授权认证、资源发现和资源访问等共同问题的解决方案, GT2 加快了网格应用的构建。GT2 通过定义和实现“标准”协议和服务, 真正地实现了可互操作的网格系统。

但是, GT2 毕竟目的是针对网格的具体实现, 它并没有一个正式的标准, 也没有接受公开的审阅, 所以在某种程度上限制了它的发展。随着网格技术的快速发展和全球网格论坛这个标准性的机构出现, 修订 Globus Toolkit 的协议的标准提到了日程上。

2.1.2 基于 OGSI 的 Globus Toolkit 3

2002 年 2 月, 在加拿大多伦多市召开的全球网格论坛 GGF 会议上, Globus 项目组和 IBM 共同倡议了一个全新的网格标准 OGSA。它把 Globus 标准与以商用为主的 Web Services 的标准结合起来, 网格服务统一以 Services 的方式对外界提供。2003 年符合 OGSA 规范的 Globus Toolkit 3.0 (GT3) 发布, 这标志着 OGSA 已经从一种理念、一种体系结构, 走到付诸实践的阶段了。

GT3 提供了一个完整的开放网格服务基础设施 (OGSI) 实现, 它的许多功能重构成与 OGSI 兼容的服务, 如服务发现、程序执行作业的提交、监控和可靠的文件传输等服务。其它如数据传递、副本定位和授权等服务也尽量构建成与 OGSI 相兼容。并且 GT3 定义了一组关于使用 WSDL 和 XML 模式的约定与扩展, 以便启用有状态服务。

虽然 OGSI 的概念很重要, 但是也存在一些自身的缺陷, 需要一些新的结构来解决这些问题。

所以采用新的结构代替 OGSi 是有必要的, 通过转变可以获得网络服务强有力的支持。

2.1.3 基于 WSRF 的 Globus Toolkit 4

2004 年 1 月, 美国 Akamai Technologies、美国 Globus Alliance、惠普、IBM、美国 Sonic Software 和美国 TIBCO Software 六公司公布了统一网格计算和 Web 服务²的新标准“WS-Notification”和“WS-Resource Framework”。Web 服务资源框架 (WSRF) 是 OGSi 的重构和发展, 利用新的 Web 服务标准。

WSRF 基本保留了 OGSi 中的所有功能, 同时更改了一些语法, 并且还在其表示中采用了不同的技术。Web 服务通知 (WSN) 为 Web 服务提供基于消息发布和预定能力。WSRF 和 WSN 都是建立在已存在的 Web 服务定义和技术基础上的, 帮助实现了网格计算、系统管理和 Web 服务的统一。

2005 年 1 月 31 日发布的 Globus Toolkit 4 (GT4), 实现了 WSRF 和 WSN 标准。GT4 提供 API 来构建有状态的 Web 服务, 其目标是建立分布式异构计算环境。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 服务的组件。

Globus toolkit 随着体系结构的发展而不断完善, 它是对体系结构的具体实现, 为构建基于 OGSA 的网格环境提供基础平台, 也为 Web 服务和网格技术的融合提供技术保障。

2.1.4 WSRF 与 OGSi 比较

随着 GT3 在实际中的应用操作, 人们发现 OGSi 与标准 Web 服务相差太远, 这是导致倾向 WSDL 的原因。一些关键的差别是: WSRF 不违背 WSDL, 可以期望有更好的工具包为 WSRF 提供支持; WSRF 不是面向对象的, WSRF 更容易混合和匹配。在 OGSi 中, 可以与服务实例协商有关它们的服务数据。在 WSRF 中, 可以与该服务协商有关它们的资源和它们的属性。如果用“服务实例”替换“资源”, 许多 OGSi 想法和模式在 WSRF 中也是可行的。WSRF 的优点在于它允许对同一个有状态资源有多个服务接口。而 OGSi 的优点在于它能为同一服务实例提供多个服务接口。如图 1 所示 (I Foster etc., 1999; Apache, 2005):

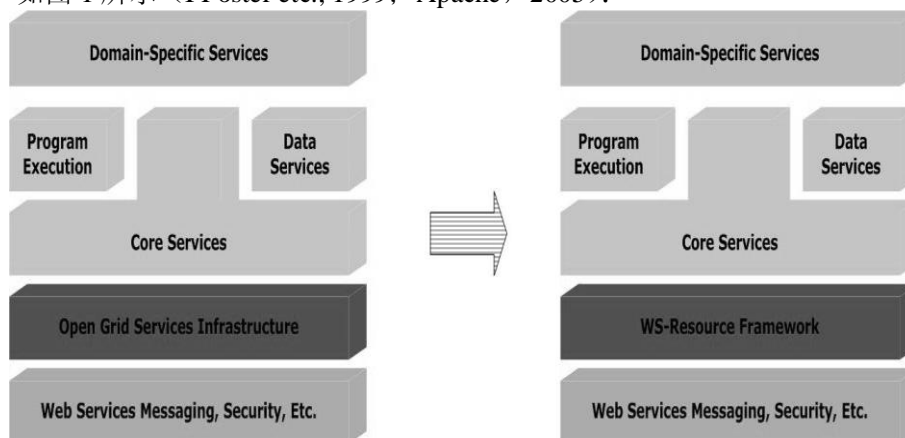


图 1 OGSi 与 WSRF 位置对照

1、WSRF 与 OGSI 的比较

WSRF 相对于 OGSI 的优点

(1) OGSI 是在一个规范中带有太多定义的重量级的规范。WSRF 把 OGSI 功能划分成为一组规范;

(2) OGSI 不能与现存的 Web 服务工具包一起很好地工作。OGSI 使用 XML 规划扩展,例如,存在时常发生的 `xsd:any`、属性使用。它也使用许多“面向文档”的 WSDL 操作。这些特征引发了一些问题,例如 JAX-RPC。WSRF 在某种程度上可减少 XML 规划的使用;

(3) OGSI 把一个有状态的资源刻画成为一个封装资源状态的 Web 服务,把服务的标示、生命周期和资源状态耦合在一起。WSRF 重新清晰地说明了基本的 OGSI 体系结构,给出在无状态 Web 服务和在其上运行的有状态资源之间的清晰的区别。

OGSI 相对 WSRF 的优点

(1) 具有面向对象范例 (OOP, Object-Oriented Paradigm) 的特征, OGSI 通过继承可获得很强的扩展性,而这在 WSRF 中是缺乏的,人们可以为开发强烈依赖 OGSI 继承的服务设想设计框架;

(2) 类似地,对 OGSI 是重量级的批判也有它的另一面:在 OGSI 中,可以对提供确定端口类型和行为的“OGSI 兼容网格服务”进行计数,但这在 WSRF 中是不容易做到的,在 WSRF 中的每一个事情都是可选择的。

2、在 WSRF 和 OGSI 中的有状态资源的模型化

WSRF 和 OGSI 两者关心的是对带有状态信息的标准 Web 服务扩展,但是两者采用不同的途径。如图 2 所示(Globus, 2006), WSRF 把 Web 服务和有联系的资源分离开来。它使用一个 WS-Addressing 端点参照来把 Web 服务和有状态的 WS-Resource(在其中采用 Reference Properties 指定资源 ID)联系起来。

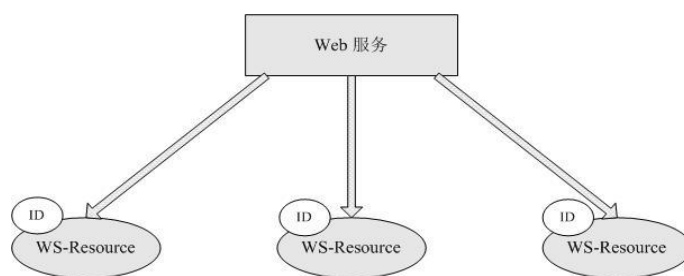


图 2 在 WSRF 中 Web 服务与有状态资源相结合

OGSI 用服务数据元素模拟有状态资源,服务数据元素与网格服务或服务实例是紧密耦合在一起的,如图 3 所示(Globus, 2006)。

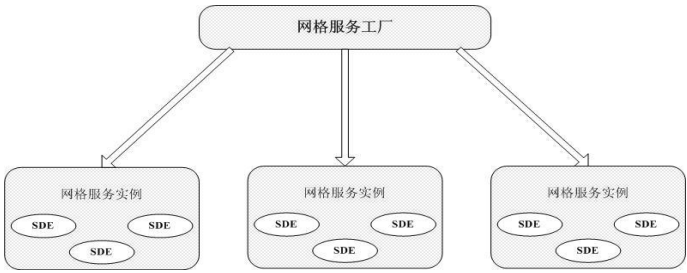


图 3 OGSI 中有状态网格服务与服务数据相结合

3、WSRF 对 OGSI/GT3 的影响

如表 1 所示 (Maozhen Li, 2006), WSRF 基本保持 OGSI 的全部概念, 引入对 OGSI 报文和它们相关语义的一些变化。对修改一个基于 OGSI 系统所需的努力或对使用 WSRF 规范的期望将是较小的。使用基于 OGSI 中间件实现的服务, 例如 GT3, 可能需要一些修改才能开发出基于 WSRF 的工具, 但是期望这种修改应该是适度的, 这应归于 WSRF 和 OGSI 之间的相似性。更一般情况是, WSRF 是基于主流 Web 服务标准并且已经被大众认可。

表 1 OGSI 和 WSRF 功能对照

OGSI	WSRF
Grid Service Reference	WS-Addressing Endpoint Reference
Grid Service Handle	WS-Renewable Reference
Handle Resolver	WS-Renewable Reference
Service Data	WS-Resource Properties
Grid Service Lifetime Management	WS-Resource Lifetime
Notification	WS-Notification
Factory	Implied WS-Resource Pattern
Service Group	WS-Service Group
Base Fault Type	WS-Base Faults

OGSI 正在被 WSRF 替代, 因为它整合了 Web 服务和网格团体两者的优势。GT3 和 GT4 不能互操作。

2.2 开放式网格体系结构 (Open Grid Services Architecture)

2.2.1 OGSA 基本思想

作为一种应用网格体系规范, 开放网格服务体系结构 OGSA(Open Grid Services Architecture) 是对五层沙漏结构的完善和发展。相对于五层沙漏结构, OGSA 强调以“服务”为中心。在五层沙漏结构中也突出了“服务”的概念, 但主要是指被共享的各种物理资源。而在 OGSA 中, 服务所指的内容非常广泛, 包括各种计算资源、存储资源、数据资源、信息资源、软件资源等, 即一切都是“服务”, 也就是“网格服务”(Grid Services)。

OGSA 符合标准的 Web 服务框架, 可以说它是网格最新研究成果与 Web Services 技术相融合的产物。在 OGSA 规范中, 网格服务本身是一种 Web 服务。该服务提供了一组被明确定义的接口, 这些接口遵守特定的惯例, 解决网格中服务的发现、服务的动态创建、生命周期管理、消

息通知机制等关键性问题。这样，网络就可以看作是具有可扩展性能的网络服务的集合。在 OGSA 应用中，网络服务能够以不同的方式聚合起来，以满足各种虚拟组织的特定需要。

2.2.2 OGSA 的主要架构分析

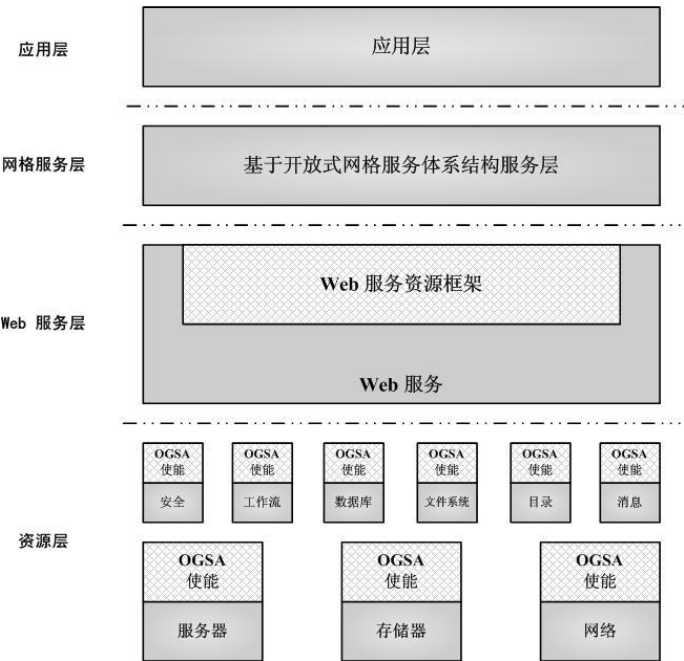


图 4 开放式网络服务体系结构图

支持 Web Services 的 OGSA 架构如图 4(IBM, 2004)。OGSA 架构由四个主要的层构成，从下到上依次为：资源层—包括物理资源和逻辑资源，Web 服务以及定义 Web 服务资源框架扩展，基于 OGSA 架构的服务，网络应用程序。下面逐一介绍这些层：

1、物理逻辑资源层

资源的概念是 OGSA 以及通常意义上的网络计算的中心部分。构成网络能力的资源并不仅限于处理器。物理资源包括服务器、存储器和网络。物理资源之上是逻辑资源。它们通过虚拟化和聚合物理层的资源来提供额外的功能。通用的中间件，比如文件系统、数据库管理员、目录和工作流管理人员，在物理网络之上提供这些抽象服务。

2、Web 服务层

OGSA 架构中的第二层是 Web 服务。一条重要的 OGSA 原则是：所有网络资源（包括逻辑的与物理的）都建模成为服务。WSRF 规范定义了网络服务，网络服务建立在标准 Web 服务技术之上。WSRF 利用 XML 与 Web 服务描述语言（WSDL，Web Services Description Language）这样的 Web 服务机制，为所有网络资源指定标准的接口、行为与交互方法。WSRF 进一步扩展了 Web 服务的定义，提供了动态的、有状态的和可管理的 Web 服务的能力，这在对网络资源进行建模时都是必需的。

3、基于 OGSA 架构的网络服务层

Web 服务层及其 WSRF 扩展为下一层提供了基础设施：基于架构的网络服务。GGF 目前正在致力于在诸如程序执行、数据服务和核心服务等领域中定义基于网络架构的服务。随着这些新

架构的服务开始出现, OGSA 将变成更加有用的面向服务的架构 (SOA)。

4、网格应用程序层

随着时间的推移, 一组丰富的基于网格架构的服务不断被开发出来, 使用一个或多个基于网格架构的服务的新网格应用程序亦将出现。这些应用程序构成了 OGSA 架构的第四个主要的层。

2.2.3 OGSA 端口类型

OGSA 提供下列可扩展的 WSDL 端口类型的接口来定义网格服务。在 OGSA 中, GridService 接口必须由所有的网格服务实现, 而其他接口是可选的。OGSA 支持下列接口:

1、GridService 端口类型

网格服务必须实现 GridService 端口类型, 因为其用做 OGSA 的基本接口。这个端口类型与在 C++和 Java 等面向对象程序设计中的基本对象类的概念是类似的, 它封装了组件模型中的根本特性。由 GridService 端口类型封装的 3 个方法是 FindServiceData ()、SetTermination Time () 和 Destroy (), 分别用于服务发现、自省和软状态生命周期管理。

2、Factory 端口类型

Factory (工厂) 是实现 Factory 端口类型的持续网格服务。它可以用于创建带有 CreateService ()、方法的短暂的网格服务实例。

3、HandleResolver 端口类型

实现 HandleResolver 端口类型的网格服务可以通过 FindbyHandle () 方法实现 GSH 到 GSR 的转变。

4、Registration 端口类型

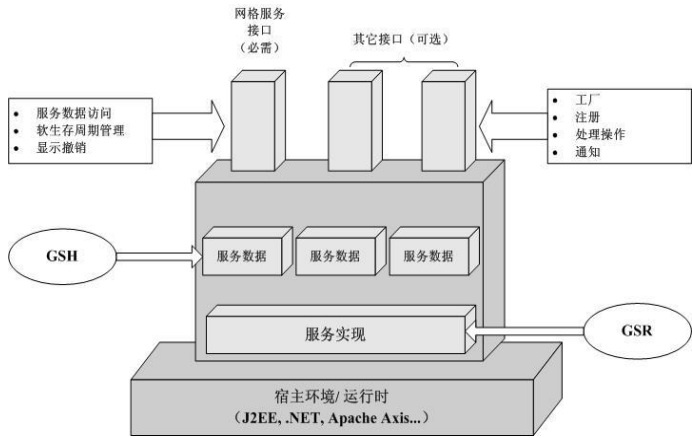
注册器是实现注册端口类型的网格服务, 通过维护 GSH 的集合和它们的有关策略提供服务发现。客户机可以通过查询一个注册器去发现可以提供的服务、属性和策略。两个元素定义一个注册服务, 即注册接口, 允许一个服务实例注册带有注册服务、相关联的服务数据集合的 GSH, 这些数据集合包含有关注册 GSH 和服务实例运行时状态的信息。RegisterService ()、UnRegisterService () 是在端口类型中为服务注册和撤销注册定义的两个方法。

5、NotificationSource / NotificationSink 端口类型

OGSA 通知模型允许有关各方订阅服务数据元素和接收通知事件 (当它们的值被修改后)。实现通知源端口类型的网格服务称为通知源。实现通知接收器端口类型的网格服务被称为通知接收器。对特定的网格服务订阅通知, 通知接收器使用在 NotificationSource 接口中的 SubscribeToNotificationTopic () 方法调用通知源, 给它通知接收器的 GSH 服务和感兴趣的主体。通知源将使用在 NotificationSink 接口中的 DeliverNotification () 方法向接收器发送一个通知报文流, 而接收器发送周期报文来通知那些对接收通知依然感兴趣的源。为确保可靠的交付, 一个用户可以定义一个绑定该服务的适当协议来实现这个行为 (性能)。

如图 5 所示(I Foster, 1998), 网格服务必须实现 GridService 接口, 而其他接口也可以实现, 但不是必须实现, 例如工厂、注册、HandleResolver 和通知源/通知接收器。OGSA 定义了服务创建、撤销、生命周期管理、服务注册、发现和服务通知等标准机制。网格服务可以是持续的服务, 也可以是短暂的服务实例。每一个网格服务都有一个唯一的 GSH 和一个或多个 GSR 去参照它的

实现，其独立于位置、平台和编程语言。一个网格服务可以部署在装有 J2EE、.Net 或 Apache Axis 主机的环境中。



2.2.4 OGSA 的服务接口

OGSA 符合标准的 Web Services 框架。Web Services 解决了发现和激活永久服务的问题，但是在网格中有大量的临时服务，因此 OGSA 对 Web Services 进行了扩展，提出了网格服务（Grid Service）的概念，使得它可以支持临时服务实例，并且能够动态创建和删除。

表 2 网格服务接口说明

PortType	操作	描述
GridService	FindServiceData	查询网格服务实例的各种信息，包括基本的内部信息、大量关于每个接口的信息以及与特定服务有关的信息。
	SetTerminationTime	设置并得到网格服务实例的终止时间。
	Destroy	终止网格服务实例。
NotificationSource	SubscribeToNotificationTopic	根据感兴趣的消息类型和内容说明，相关事件的通知发送者进行登记。
	UnSubscribeToNotificationTopic	取消登记。
NotificationSink	DeliverNotification	异步发送消息。
Registry	RegisterService	网格服务句柄的软状态注册。
	UnRegisterService	取消注册的网格服务句柄。
Factory	CreateService	创建新的网格服务实例。
PrimaryKey	FindByPrimaryKey	返回根据特定键值创建的网格服务句柄。
	DestroyByPrimaryKey	撤销特定键值创建的网格服务实例。
HandleMap	FindByHandle	返回与网格服务句柄相联系的网格服务实例。

表 2 列出了网格服务的接口 (Maozhen Li, 2006), 其中只有 GridService 接口是必须的, 而其他的接口都是可选的。每个接口定义了一些操作, 这些操作通过交换定义好的一系列消息来激活。网格服务接口和 WSDL 的 portTypes 相对应, 网格服务提供 portTypes 的集合, 包括一些与版本有关的附加信息, 在网格服务中用 serviceType 来描述, serviceType 是 OGSA 定义的 WSDL 的扩展元素。

2.3 OGSA-DAI

OGSA-DAI 项目的主要目标是致力于构建一个支持访问大型的、基础的静态数据库的中间件系统, 这个中间件是带有许多扩展点的工具包, 它为开发者提供了便利, 以适应各自特定的不同的需要。OGSA-DAI 为 OGSA 规范提供扩展, 允许数据源, 如数据库, 在一个 OGSA 框架中被访问。通过 OGSA-DAI 接口所能达到的最终目标是, 分散的, 异构的数据源能够像单一逻辑的数据源一样被访问和控制。OGSA-DAI 组件提供的潜在在构建高级服务中可以作为基础组件, 支持在虚拟组织中的数据集成和分布式查询处理。



图 6 OGSA-DAI 在 OGSA 中的位置

2.3.1 OGSA-DAI 基本思想

OGSA-DAI 是一个网格中间件, 设计用于简化对数据网格中的数据进行访问和集成。OGSA-DAI 的主要开发是由苏格兰的 EPCC 和英格兰的 IBM Hursley 进行的。OGSA-DAI 的催化剂是大量“大型科学”项目, 例如 AstroGrid、eDiaMoND、Datamining Grid 和 myGrid, 需要中间件层提供对大型数据库的访问。

OGSA-DAI 的结构像是一个工具包, 并且带有一些扩展点, 让开发人员可以扩充自己的功能来满足特定的需要。它通过基于 Web 服务的接口来提供数据资源, 这样就可以简单地将数据源作为网格中的首要元素进行操作。目前所支持的数据源有关系数据库 (通过 JDBC)、XML 数据库 (通过 XMLDB) 和文件系统。当有新的需求出现时, 应用程序开发人员也可以编写自己的数据库支持程序。

OGSA-DAI 提供了以下主要功能:

(1) 可以提供数据源。可以对关系型数据源执行 SQL 查询, 对 XML 集执行 XPath 语句, 或者通过 Web 服务调用对大文件进行搜索查询。

(2) 可以在服务中实现其他功能。数据源之外进行的数据转换可以通过一些可以避免不必要的数据移动的操作实现。

(3) 提供了一种紧凑的方法在单个请求中处理与某个服务的多次交互。这可以通过 XML 文档(称为 执行文档)来实现。当数据流进入某个数据源或从某个数据源检索出来时, 对其进行的操作之间是通过管道来传递数据的。

(4) 使开发人员可以简单地在 OGSA-DAI 中添加或扩充功能——执行文档和底层的框架都可以进行扩充, 这样就可以添加新功能, 或者对现有的功能进行定制。

(5) 可以通过服务接口查询有关数据和数据源的元数据。

2.3.2 OGSA-DAI 的数据服务结构

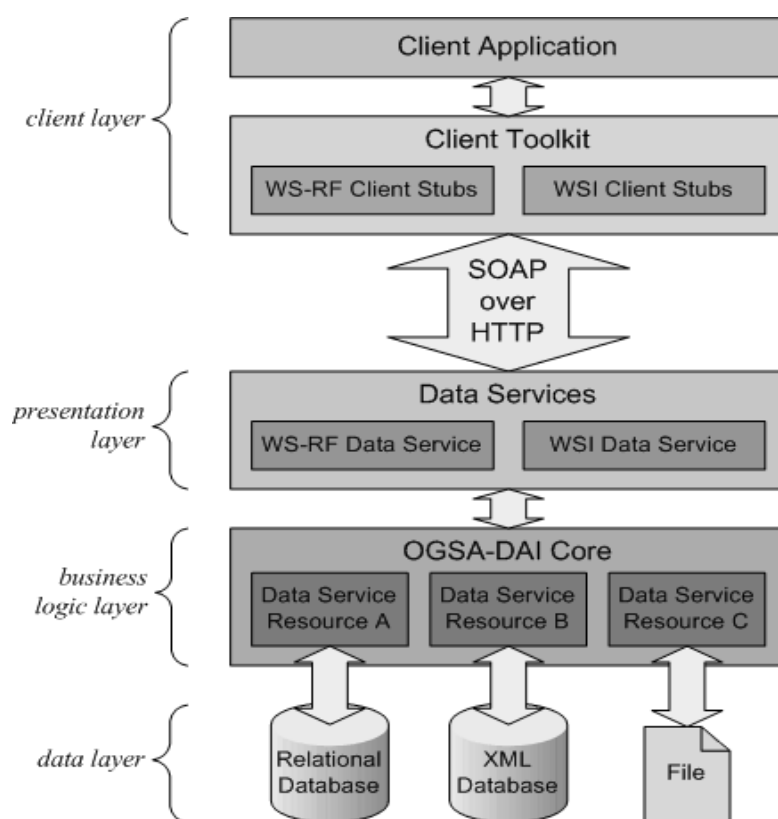


图 7 OGSA-DAI 结构的总体示意图

OGSA-DAI 的数据服务结构如图 7 所示 (IBM, 2005)。结构中包含许多层, 数据层、事务逻辑层、表示层和客户层。每层均有各自的功能和服务目的。从底层向上依次为: 数据层、事务逻辑层、表示层和客户层。各层组件和各接口功能说明如下:

1、数据层

数据层由一系列数据源组成, 这些数据源可通过 OGSA-DAI 被体现。目前, 这些数据源包括: 第一, 关系型数据库: MySQL、SQL Server、DB2、Oracle、PostgreSQL。

第二, XML 数据库: eXist、Xindice。

第三, 文件系统: OMIM、SWISSPROT、 EMBL 等。

2、事务逻辑层

事务逻辑层概括了 OGSA-DAI 的核心功能, 它由一系列被称为数据服务资源的组件构成。如图 7 所示, 部署多种数据服务资源用以体现多种数据源。并且, 在数据服务资源与数据源之间存在着的一对一关系。

3、表示层

表示层概括了通过 web 服务接口体现数据服务资源的功能要求。OGSA-DAI 包括两种实现途径: WSRF 数据服务和 WSI 数据服务。每种实现方法都会用到 WSDL 文档描述接口。

4、客户层

客户端通过相应的数据服务实现与数据服务资源的交互, 根据 WSRF 或 WSI 数据服务的不同部署, 要求客户应用层与 WSRF 或 WSI 标准一致。

2.3.3 OGSA-DAI 端口类型

OGSA-DAI 提供下列接口(端口类型)定义用于数据访问和集成的网格服务。在 OGSA-DAI 中的网格服务称为网格数据服务(GDS, Grid Data Service)。GDS 必须实现 GDSPortType 接口。OGSA-DAI 提供了下列端口(W Allcock,etc. 2000; GGF OGSA-DAI Group, 2006)。

1、GDSPortType 端口类型

GDSPortType 端口类型支持数据存取、集成和交付, 并且所有的 GDS 要实现该端口类型。它扩展了 3 个端口类型。一个是 GridService 端口类型, 另外两个是由 OGSA-DAI 定义的 GridDataTransport 端口类型。

2、GridDataPerform 端口类型

GridDataPerform 端口类型为客户端提供了访问数据源和找回结果的方法。如图 8 所示, 它支持用于数据库查询的面向文档的接口, 其中查询请求是通过使用网格数据服务执行(GDS-Perform)文档来提交以指定数据源上的操作, 而相应是通过使用包含操作结果的网格数据服务相应(GDS-Response)文档来返回。提交给 GDS 的查询文档的特性和由此产生的结果文档, 取决于给出的网格服务配置的数据源的类型。例如, 关系数据库可以接收 SQL 查询, 而 XML 数据库可以接收 XPath 查询。使用一个文档来描述请求允许对请求进行分析和优化。

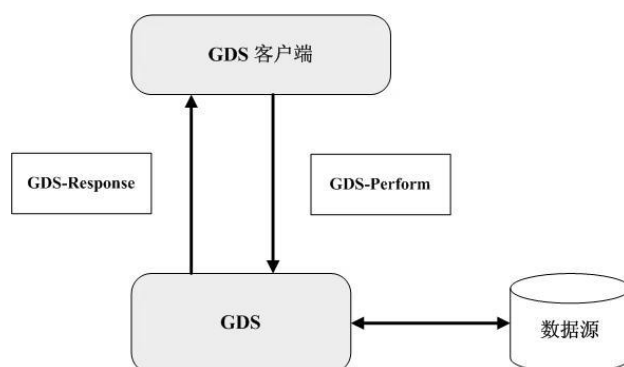


图 8 访问 OGSA-DAI 中的数据

3、GridDataTransport 端口类型

GridDataTransport 端口类型提供在 OGSA-DAI 服务之间，以及在 OGSA-DAI 客户机和 OGSA-DAI 服务器之间数据传输的支持。它允许抽取或推送数据。

PutFully () ——传输完整的数据集合。

GetFully () ——接收完整的数据集合。

PutBlock () ——传输数据块，该数据块是使用指定块索引的操作的一大批数据的一部分。

GetBlock () ——接收数据块，该数据块是使用指定块索引的操作的一大批数据的一部分。

4、GridDataServiceFactory 端口类型

GridDataServiceFactory 端口类型用于实现网格数据服务工厂 GDSF，是一个用于创建 GDS 的持续网格数据服务。该端口类型扩展了 3 个端口类型：GridService、Factory、NotificationSource。

5、DAIServiceGroupRegistry 端口类型

DAIServiceGroupRegistry 端口类型用于实现 DAIServiceGroupRegistry (DAISGR)。一个 DAISGR 服务能够用于注册实现一个或多个 OGSA-DAI 端口类型的任意服务。DAIServiceGroupRegistry 端口类型扩展了 4 个端口类型：GridService、ServiceGroup、Registration 和 NotificationSource。

2.3.4 OGSA-DAI 支持的数据库

OGSA-DAI 支持的数据库类型(Ian Foster, etc. 2002):

表 3 OGSA-DAI 支持数据库类型说明

数据库名称	版本	主页
MySQL	3.2.3 及以上	http://www.mysql.com/downloads/
IBM DB2	-	http://www-306.ibm.com/software/data/db2/
注：OGSA-DAI 不支持 DB2 中对数据库的创建和删除操作。		
Microsoft SQL Server	-	http://www.microsoft.com/sql/
注：OGSA-DAI 不支持 SQL Server 图片及多媒体功能。		
Oracle	10g Enterprise Edition Release 及以上	http://www.oracle.com/database/
注：OGSA-DAI 不支持 Oracle 中对数据库的创建和删除操作。		
注：Oracle 数据库的 JDBC 并不支持标准 JDBC 的 BLOB 插入操作，因此 OGSA-DAI 目前也不支持 Oracle 中对表空间的 BLOB 插入操作。		
PostgreSQL	-	http://www.postgresql.org/
XML		
eXist	Build from 2005-12-03	http://exist.sourceforge.net/
注：eXist 运行在 OGSA-DAI 或其它网格中间件（例如 Globus Toolkit）时，会发生内存泄露。		
Files		
File system	-	-

其他的数据库类型:

数据库名称	版本	主页	JARs	类名
HSQL	1.7.1	http://hsqldb.sourceforge.net 注: 仅进行小范围试验。	hsqldb.jar	org.hsqldb.jdbcDriver
XML				
Apache Xindice	1.1	http://xml.apache.org/xindice/ 注: 仅进行小范围试验。	xindice.jar	org.exist.xmlldb.DatabaseImpl

2.4 Web Services

2.4.1 Web 服务的定义及特征

1、Web Services 定义

Web 服务是一种被设计成为支持机器之间通过互联网实现交互操作的软件系统,它具有一种能够使用机器可处理格式(尤其是 WSDL)描述的接口,其他系统按照 Web 服务描述信息中指定的方式使用 SOAP 消息与之交互,典型的是使用具有 XML 序列化特征的 HTTP 进行传输,并结合使用其他相关 Web 标准。

2、Web Services 特征

从上述定义可知,Web 服务使用标准的互联网协议,像超文本传输协议 HTTP 和 XML,将功能体现在互联网和企业内部网上。

Web 服务是一种部署在 Web 上的对象/组件,是下一代分布式系统的核心,它具备以下特征(W3C, 2006):

(1) 完好的封装性: Web 服务既然是一种部署在 Web 上的对象,自然具备对象的良好封装性。从封装的粒度来看,Web 服务一般封装了一个离散的(单独的)功能,一个 Web Services 完成一个独立的任务。对于使用者而言,他仅能看到该服务的描述。

(2) 松散耦合: 这一特征也是源于对象/组件技术,当 Web 服务的实现发生变更的时候,调用者是不会感到这一点的,对于调用者来说,只要 Web 服务的调用接口不变,Web 服务的实现发生任何变更对他们来说都是透明的,甚至是当 Web 服务的实现平台从 J2EE 迁移到了 .NET 或者是相反的迁移流程,用户都可以对此一无所知。对于松散耦合而言,尤其是在 Internet 环境下的 Web 服务而言,需要有一种适合 Internet 环境的消息交换协议。而 XML/SOAP 正是目前最为适合的消息交换协议。

(3) 使用协议的规范性: 这一特征从对象而来,但相比一般对象其界面更加规范化且易于机器理解。首先,作为 Web 服务,对象界面所提供的功能应当使用标准的描述语言来描述(比如 WSDL),而由标准描述语言描述的服务界面应当是能够被发现的,因此这一描述文档需要被存储在私有的或公共的注册库里面;其次,由于安全机制对于松散耦合的对象环境的重要性,我们需要对诸如授权认证、数据完整性(比如签名机制)、消息源认证以及事务的不可否认性等运用规范的方法来描述、传输和交换;最后,在所有层次的处理都应当是可管理的,因此需要对管理协议运用同样的机制。

(4) 使用标准协议规范: 作为 Web 服务,其所有公共的协议完全需要使用开放的标准协议

进行描述、传输和交换。这些标准协议具有完全免费的规范，以便由任意方进行实现。一般而言，绝大多数规范将最终有 W3C 或 OASIS 作为最终版本的发布方和维护方。

(5) 高度可集成能力：由于 Web 服务采取简单的、易理解的标准 Web 协议作为组件界面描述和协同描述规范，完全屏蔽了不同软件平台的差异，无论是 CORBA、DCOM 还是 EJB 都可以通过这一种标准的协议进行互操作，实现了在当前环境下最高的可集成性。

(6) 开放性：Web Services 可以与其他 Web Services 进行交互。它具有语言 and 平台无关性。支持 CORBA、EJB、DCOM 等多种组件标准。支持各种通讯媒体如：HTTP、SMTP、MQ、FTP 等。

2.4.2 Web 服务的工作原理

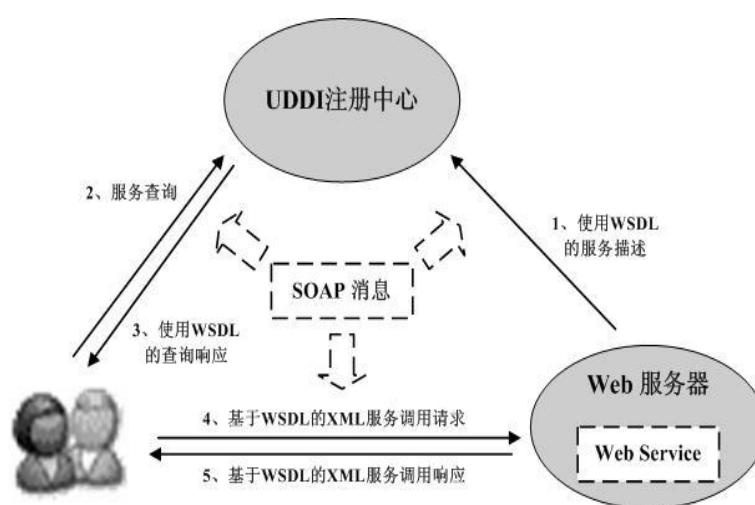


图 9 Web 服务工作原理

在通过基于服务角色和服务协议栈介绍了 Web 服务体系结构之后，下面主要说明 Web 服务的工作原理，如图 9 所示（W3C, 2005; Sandholm, etc. 2006），也就是 Web 服务描述、发布、查询、绑定、调用的整个流程：

第一步：服务提供者在开发完 Web 服务后，使用 WSDL 对服务的相关信息进行描述，并在 UDDI 注册中心发布该服务；

第二步：服务请求者，向 UDDI 注册中心发送该服务的查询请求；

第三步：注册中心返回查询响应信息，如果服务在注册中心中存在，则返回服务的描述信息；

第四步：如果服务请求者获得服务的描述信息，则根据描述信息设置相关服务调用参数，并向查询到的服务提供者发送服务调用请求；

第五步：服务提供者接收到服务请求者的服务调用请求后，解析请求信息，并根据参数调用服务，将处理结果以 XML 形式返回服务请求者。服务请求者获得服务调用结果后，便完成整个 Web 服务操作流程。

第三章 基于 OGSA 农业科技数据库集成 (AST-DB) 设计

3.1 AST-DB 集成的作业流程设计

农业科技数据库本身数据庞大、系统复杂,我们所期望的是科技用户通过简易的访问流程实现数据服务。基于 OGSA 的 AST-DB 系统中,农业科技用户只需使用资源描述语言(RSL, Resource Specification Language)说明运行作业,即能指出如可执行名称、输入/输出作业目录以及运行队列等。

AST-DB 集成系统仍旧延续 Globus Toolkit 所提供的运行远程作业的方式,使用一系列 WSDL 文档和用户端接口提交、监控和终止作业。作业管理执行与作业管理器的创建一一对应。但是,这种服务必然会带来很多问题,问题之一即为,作业管理器中需要对每一个用户作业进行管理,但当用户不使用资源时,工作处于空闲状态,但这些工厂仍处于资源消耗状态。

为此,基于 OGSA 的 AST-DB 集成作业流程设计,引入代理服务机制,即服务器端代理服务和用户端代理服务。服务器端代理服务拥有公共管理机制作用,通过它可以调用用户端数据资源,为用户创建所需要的作业管理服务。在服务请求发生时,如果存在用户端代理服务,则代理路由器会路由一个用户引入的请求到其他用户端;如果不存在,则可以为用户标记请求,路由到服务器端数据管理服务。如图 10 所示(Maozhen Li etc., 2006; Papakonstantinou Y, etc.,1995),服务器端数据资源、服务器端代理服务、用户端数据资源、用户端代理服务和作业管理器之间存在的关系。每一个用户端数据管理服务运行在对应的一个用户端运行环境,并且在该用户端运行环境的生命周期内是有效的。每个用户端可以创建一个或多个作业管理器。

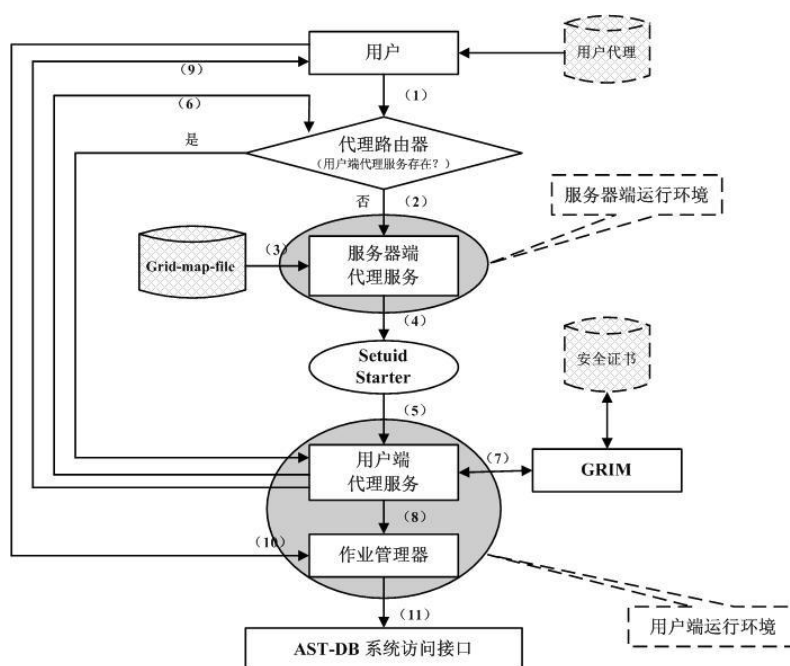


图 10 基于 OGSA 的 AST-DB 集成作业流程

每一个活动的用户有与之对应的运行环境,一个用户端代理服务和一个或多个作业管理器。

这种方法允许以轻量级的方式创建多个服务。

(1) 用户用 **RSL** 构成作业描述并用 **GSI** 证书签署, 签名的请求被发送到代理路由器。

(2) 代理路由器接受请求, 并路由该请求到用户端代理服务 **A** (如果存在的话转到 (6))。如果满足请求, 则返回相应数据资源; 如果不满足请求, 则代理路由器会将请求路由到服务器端代理服务或其他用户端代理服务 (注: 此功能涉及到网格技术资源调度问题, 在此只做假设运用)。

(3) 服务器端代理服务验证请求的签名并建立用户标识。然后它基于用户标识和网格映射文件 (**grid-map-file**) 来确定作业将被执行的用户。**grid-map-file** 是包含从 **GSI** 身份到用户身份映射的用户端配置文件。

(4) 服务器端代理服务调用 **setuid Starter** 进程, 为请求者开始用户端代理服务 **B**。

(5) **setuid Starter** 是一个特权程序 (典型的 **setuid-root**), 具有为用户启动预先配置的用户端代理服务的单一功能。

(6) 当用户端代理服务 **B** 启动后, 该服务需要获得证书并向代理路由器注册。为了注册, 用户端代理服务 **B** 发送报文给代理路由器, 通知代理路由器用户端代理服务 **B** 的存在, 这样代理路由器可以路由将来的作业初始化请求给用户端代理服务 **B**。

(7) 用户端代理服务 **B** 调用网格资源身份映射 (**GRIM**, **Grid Resource Identity Mapper**) 来获得一系列的证书。**GRIM** 是特权程序 (典型的 **setuid-root**), 访问用户端证书并通过它们为用户端数据管理服务生成一套 **GSI** 代理证书。这些代理证书³已经被嵌入到用户网格身份、本地账户名字和策略中, 这可以帮助客户机验证用户端代理服务是否是所需的用户端代理服务。

(8) 用户端代理服务 **B** 接收已签署的作业请求。用户端代理服务 **B** 验证在请求上的签名, 确认它没有被篡改, 并验证请求者是访问正在用户端代理服务 **B** 运行的用户的授权者。一旦这些验证完成, 用户端代理服务 **B** 创建并调用一个带作业初始化请求的管理作业服务。

(9) 用户端代理服务 **B** 返回管理作业服务的地址 (**GSH**) 给用户。

(10) 用户连接到管理作业服务, 去初始化作业提交。请求者和管理作业服务使用从 **GRIM** 获得的证书执行相互授权。管理作业服务验证请求者是否是在本地账户中初始化进程的授权者。请求者用 **GRIM** 证书 (从一个适当的、包含用户的网格标识主机证书产生而来) 授权管理作业服务。这个方案允许用户端去验证管理作业服务正在服务于它通信, 然后用户委派 **GSI** 证书给管理作业服务以分派作业运行。

(11) 按照用户请求, 请求并获取相应的 **AST-DB** 数据资源。

3.2 AST-DB 集成的编程设计

基于 OGSA 的 **AST-DB** 编程, 主要利用 **Globus Toolkit** 基础平台中松散耦合的用户端占位程序 (占位程序, 又称为客户端占位程序, 主要是使服务器组件看起来像是驻留在客户端计算机上) 和 **AST-DB** 服务器。**AST-DB** 集成的编程设计功能描述如下:

(1) 基于 OGSA 的 **AST-DB** 集成设计利用 **WSDL** 定义数据库的服务接口。**WSDL** 是一个具有所有 **WSRF** 定义类型、报文、端口类型和命名空间的 **WSDL**。**WSDL** 使用 `<wsdl: portType>` 标签替代在标准 **WSDL** 中的 `<portType>` 标签。

(2) 使用 **Web** 服务部署描述符 (**WSDD**, **Web Services Deployment Descriptor**) 建立及部署

数据库网格服务有关的信息,例如数据库网格服务的名字、数据库网格服务实例的名字,以及数据库网格服务实例的基本类。

(3) 基于 OGSA 的 AST-DB 集成设计将所有编译 Java 代码连同相关文件(例如数据库网格服务的 WSDD 文件)打包为一个 GAR 文件以方便服务的部署。GAR 是一个特殊种类的 JAR。

(4) AST-DB 网格服务可以是能动态地创建和显式地撤销的瞬时数据服务。

(5) 基于 OGSA 的 AST-DB 网格服务是与服务数据相联系的有状态服务。

(6) 该网格服务可以通知对感兴趣的农业科技数据库有订阅的用户。

如图 11 所示,为在实现基于 OGSA 的 AST-DB 集成应用中的数据流控制。用宿主环境 Apache Axis 实现 AST-DB 网格数据库集成应用步骤如下图所示 (Kunszt, Peter Z etc., 2002; Maozhen Li etc., 2006):

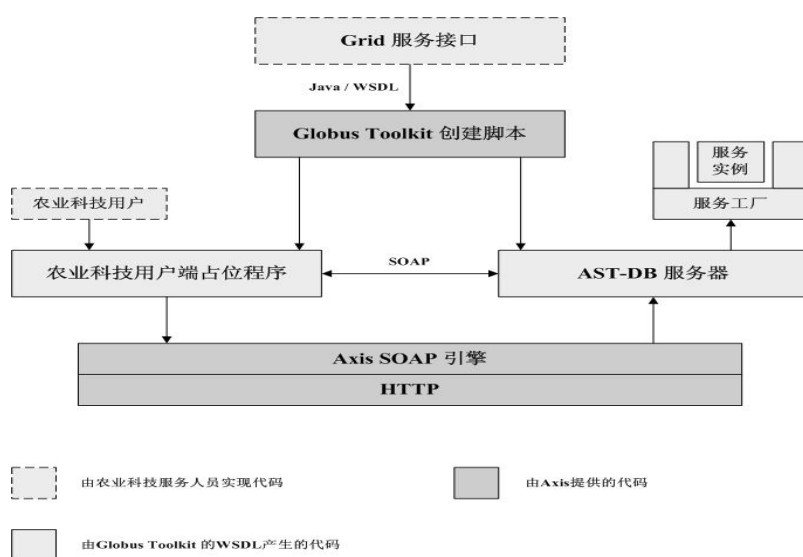


图 11 实现基于 OGSA 的 AST-DB 集成应用中的数据流控制

- (1) 编写 AST-DB 网格服务接口,该接口可用 Java、WSDL 或 GWSDL 定义;
- (2) 编写实现 AST-DB 网格服务接口的服务;
- (3) 为 AST-DB 网格服务工厂的部署编写 WSDD 文件;
- (4) 使用 Globus Toolkit 创建 AST-DB 脚本编译网格服务接口文件和实现文件,并把它们连同其他相关文件(例如从接口产生的端占位程序和 WSDD 文件)打包成为一个 GAR 文件;
- (5) 使用 Apache Ant 把 GAR 文件部署到 AST-DB 网格服务容器中以发布该服务;
- (6) 为农业科技用户编码以请求工厂,创建该服务的实例。农业科技用户首先得到 AST-DB 网格服务工厂的 GSH, 然后是工厂的 GSR。随后使用工厂的 GSR 创建实例并获得创建实例的 GSR。最后,农业科技用户该 GSR 访问该服务实例;
- (7) 启动 AST-DB 网格服务容器。
- (8) 启动农业科技用户请求服务。

3.2.1 AST-DB 集成中的服务器端组件

如图 12 所示, AST-DB 服务器端框架的主要体系结构组件包括下列内容 (J.A. Senn, 2005):

(1) Web 服务引擎: 该引擎用于处理通常 Web 服务行为、SOAP 报文处理、JAX-RPC 句柄进程和 Web 服务配置。

(2) AST-DB 网格服务容器: 基于 OGSA 的 AST-DB 为有状态网格服务的创建、撤销和生命周期管理提供一个容器。

基于 OGSA 的 AST-DB 使用 Apache Axis 作为 SOAP 引擎, 它可部署在 Tomcat Web 服务器中作为一个 Java Servlet 容器来运作。SOAP 引擎服务 SOAP 请求/相应得序列化和非序列化、JAX-RPC 句柄引用和网格服务配置。基于 OGSA 的 AST-DB 容器提供对 Axis 框架的重点处理, 向 AST-DB 网格服务容器传递请求报文。一旦 AST-DB 网格工厂创建服务实例, 该框架就会为该实例创建唯一的 GSH, 该实例通过容器注册号被注册。这个注册保持所有的有状态服务实例, 并且通过与其它的组件和处理者联系来实现服务:

第一, 标识 AST-DB 服务和调用 AST-DB 服务的方法;

第二, 得到/设置 AST-DB 服务属性 (例如实例 GSH 和 GSR);

第三, 激活/解除 AST-DB 服务;

第四, 解析 GSH 到 GSR。

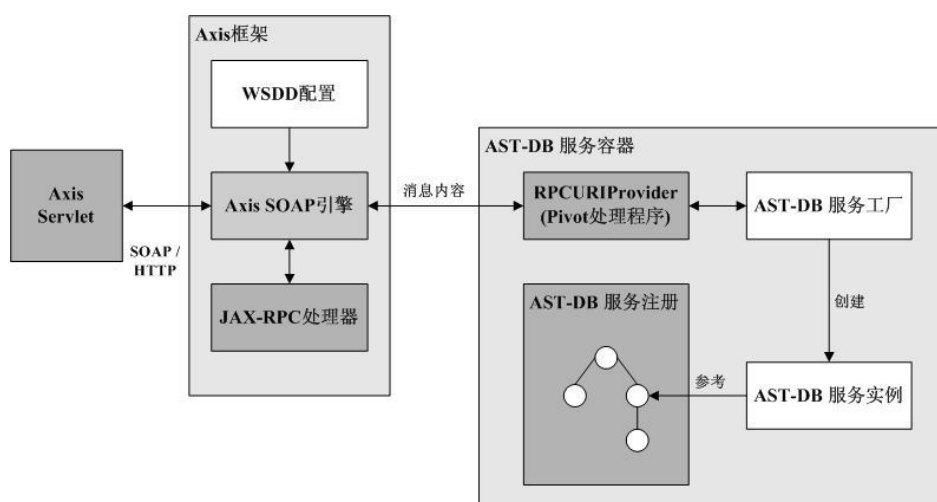


图 12 基于 OGSA 的 AST-DB 的服务端组件

3.2.2 AST-DB 集成中的客户端组件

如图 13 所示, 基于 OGSA 的 AST-DB 为网格服务 AST-DB 用户端提供通常的 JAX-RP 科技用户机端编程设计。此外, 基于 OGSA 的 AST-DB 为用户机端提供许多的帮助类, 以隐藏 WSRF 科技用户端编程设计的细节。

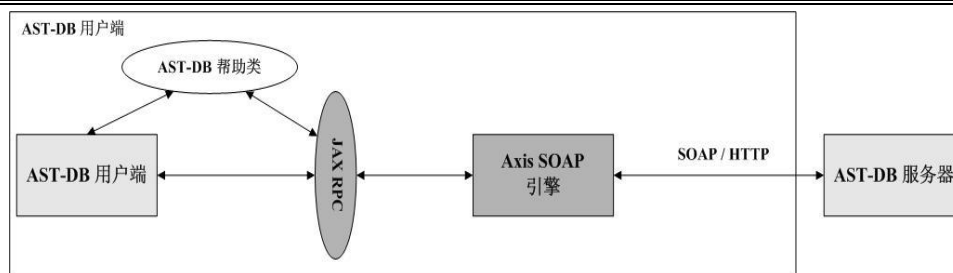


图 13 基于 OGSA 的 AST-DB 的客户端组件

3.2.3 AST-DB 编程设计小结

基于 OGSA 的 AST-DB 编程设计包括 WSRF 的实现，并且已经用于在 OGSA 上下文环境中建立网格应用。基于 OGSA 的 AST-DB 编程设计有涉及核心和基本两类服务。核心服务与有状态网格服务的创建、撤销和生命周期管理相联系。基本服务安全作业提交、信息服务和可靠数据传输相联系。为了提交并行运行的作业，AST-DB 提供两种选择：

- (1) 在多个节点上部署 AST-DB 服务工厂，并以并行方式运行多个服务实例。
- (2) 使用服务器端代理服务和用户端代理服务向后端调度系统提交用户作业。

3.3 AST-DB 集成的功能设计

基于 OGSA 的农业科技数据库集成设计中引入众多的端口类型，它为基于网格的农业科技数据访问和集成提供下列功能（IBM，2006）：

3.3.1 AST-DB 生命期管理功能

农业科技数据服务是一个短暂服务。它由与数据源相联系的 GDSF 创建。科技人员可以通过与农业科技数据服务的交互来访问数据源。农业科技数据服务可以被动态地创建和显示地撤销。

3.3.2 AST-DB 服务注册 / 取消注册功能

农业科技数据库集成服务可以通过 `ServiceGroupRegistration::ADD()` 方法向 DAISGR（`DAIServiceGroupRegistry`）注册自己。除此之外，`DAISGR_A` 也能够向 `DAISGR_B` 中注册。通过查询 DAISGR，科技人员可以发现提供特殊服务或管理特殊数据元的 OGSA-DAI 服务。在 DAISGR 中注册的服务也能通过 `ServiceGroupRegistration::Remove()` 方法撤销注册。

3.3.3 AST-DB 服务发现功能

农业科技人员可以通过 `GridService::FindServiceData()` 方法查询 DAISGR 以发现满足它需要的 OGSA-DAI 服务。农业科技用户端可以直接查询该 OGSA-DAI 服务，并且也可查询管理 OGSA-DAI 服务注册的 WSRF `ServiceGroupEntry` 服务。用户端查询 DAISGR 有 3 个目的：

- (1) 发现 GDSF, 为特定的应用创建 GDS 实例;
- (2) 查询 OGSA-DAI 服务实例以获得它的状态信息;
- (3) 查询与 OGSA-DAI 服务相联系的 WSRF ServiceGroupEntry, 以找回 OGSA-DAI 服务的注册;

3.3.4 AST-DB 服务通知功能

农业科技人员向 DAISGR (DAI Service Group Registry, 数据访问集成服务注册) 订阅事件通知, 例如 DAISGR 状态的变化、新数据库服务的注册或现存服务的撤销注册。如图 14 (IFoster, 2001) 表示用于通知的步骤:

- (1) 农业科技用户使用 NotificationSource::Subscribe() 方法向 DAISGR 订阅感兴趣的服务;
- (2) DAISGR 创建订阅服务用于实现 NotificationSubscription 接口, 以管理订阅;
- (3) DAISGR 返回农业科技用户端实现 NotificationSubscription 订阅服务的标识。
- (4) 用户端通过用户标识并采用 GridService::FindServiceData() 方法查询订阅服务, 以管理用户端订阅服务, 例如生命周期管理。
- (5) 一旦 DAISGR 发生变化, DAISGR 会通过 NotificationSink::DeliverNotification() 方法将消息传送给起到中转服务作用的通知接收聚合器。
- (6) 通知接受聚合器将消息发送给农业科技用户。

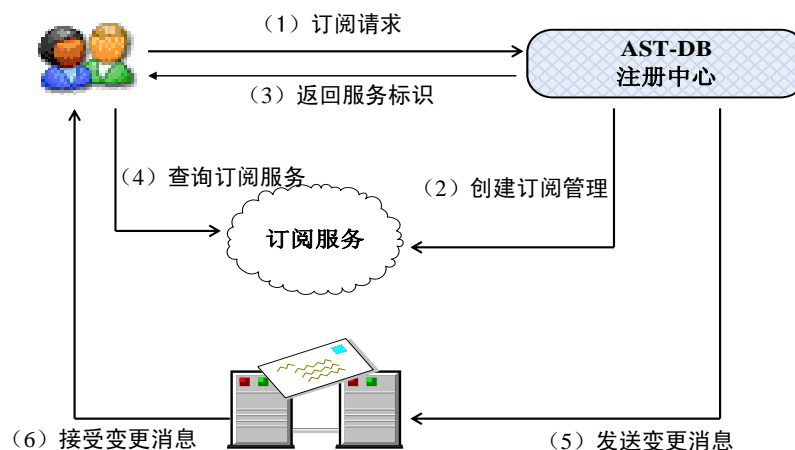


图 14 AST-DB 中的服务通知

3.4 AST-DB 的数据服务交互

现在给出在基于 OGSA 的农业科技数据库集成服务之间交互的视图, 如图 15 所示(Globus, 2004; Maozhen Li, etc. 2006)。

- (1) 启动网格服务容器, 读 serverconfig.wsdd 文件。这个 serverconfig.wsdd 文件允许数据库网格服务容器访问已部署的农业科技数据库服务的信息, 以及在服务名和相联系的类之间的映射。

- (2) 农业科技数据库网格服务容器基于在 serviceconfig.wsdd 文件中指定的 GSH 来创建持续的 DAISGR。

(3) 农业科技数据库网格服务容器基于在 serviceconfig.wsdd 文件中指定的 GSH 来创建持续的 GDSF。

(4) GDSF 用 ServiceGroupRegistration::Add() 方法在 DAISGR 注册自己。

(5) 科技用户使用 GridService::FindServiceData() 方法查询 DAISGR 科技用户选择一个已注册的 GDSF。

(6) DAISGR 返回所选择的 GDSF 的 GSH。

(7) 科技用户可以查询 GDSF 的服务数据元素，以获得它的配置信息。

(8) 科技用户调用 GDSF Factory::createService() 方法创建 GDS 实例。

(9) GDSF 创建 GDS 实例。

(10) GDSF 向科技用户返回 GDS 实例的 GSH。

(11) 科技数据库终端用户使用 GridService::FindServiceData() 方法查询新产生的 GDS 实例的服务数据元素，来建立它的配置和描述 GDS-Perform 文档的计划⁴，该文档可以通过 GridDataPerform::perform() 方法提交。

(12) 科技用户向 GDS 实例提交 GDS-Perform 文档。

(13) GDS 实例访问农业科技数据库去获得数据，并产生一个 GDS-Response 文档。

(14) GDS 实例向科技用户返回 GDS-Response 文档。

(15) 科技用户通过 GridService::Destroy() 方法撤销 GDS 实例。

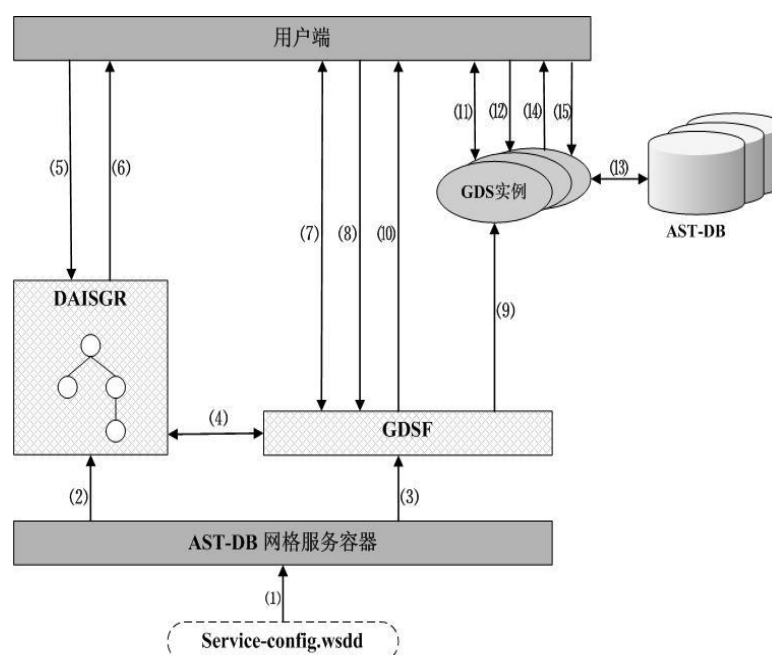


图 15 基于 OGSA 的农业科技数据库集成服务之间交互

3.5 AST-DB 集成的应用架构设计

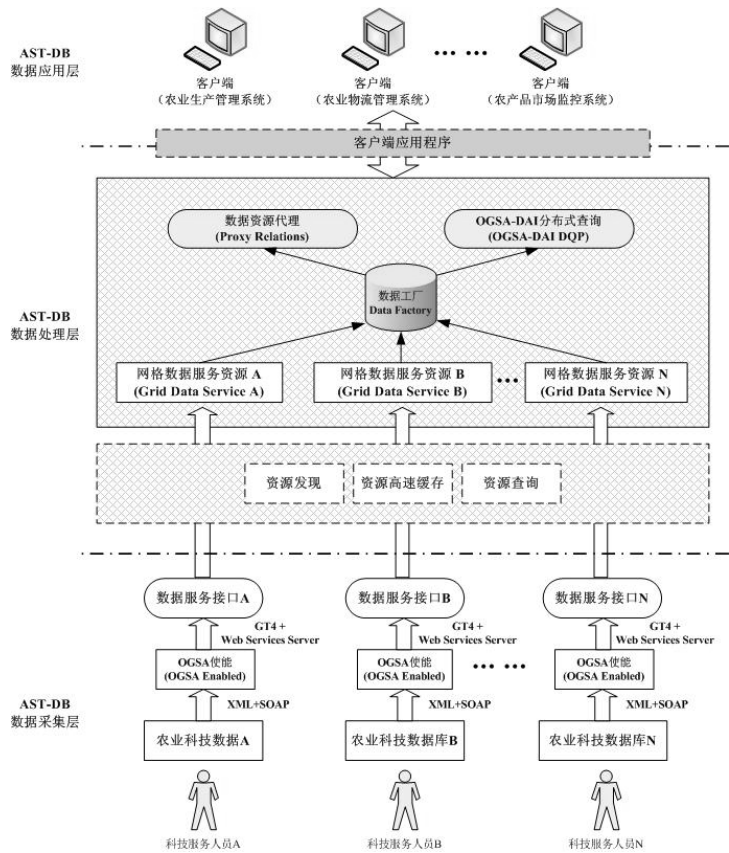


图 16 基于 OGSA 的农业科技数据库 (AST-DB) 集成的应用架构示意图

通过引入数据网格技术，尤其是 OGSA-DAI（开放网格服务架构—数据访问与集成），完成与设备接口集成，实现对农业科技数据库网格的多网络、跨平台的信息集成。体系结构主要分为三个部分，分别是 AST-DB 数据采集层、AST-DB 数据处理层和 AST-DB 数据应用层，其体系结构如图 16 所示。

3.5.1 AST-DB 数据采集层

在硬件系统方面，包括科技工作人员的存取设备，负责存储农业科技数据；在网格系统方面，将数据以规格化的 XML 文档形式描述，并通过 GT4(Globus Toolkit 4.0)实现网格环境中的 OGSA 使能(OGSA Enabled)，将农业科技数据最终转换为数据网格中可以访问的 XML 数据资源。

3.5.2 AST-DB 数据处理层

该层利用 OGSA-DAI 并发处理数据采集层所传送的事件与数据流，并利用网格数据服务对数据进行过滤和聚合，并存储到信息数据库中，以数据库 Web Services 的方式在数据网格中发布，以此作为信息资源。此外，当科技用户端数据发生变化时，农业科技数据处理层可以触发采用

OGSA-DAI 的事件机制与消息发布与订阅机制统一处理事件，以网格消息的方式通知 (Notify) 系统内其它组件，也可根据具体的规则来形成数据过滤器，用来向系统订阅数据服务。

3.5.3 AST-DB 数据应用层

该层直接面向农业科技数据流通环节的双向交流信息层。将数据处理层抽取的信息分配到各应用层管理数据库中，同时也将各管理数据库采集信息反馈到信息处理层。这样，无论是对于确定农业科技数据使用，还是对于农业科技数据的修改，均能确保实现信息的对称交流。

第四章 AST-DB 农产品价格数据集成系统研究与设计

4.1 农产品价格数据 AST-DB 集成概要设计

随着农业与农村经济的迅猛发展,农产品市场在其中发挥着越来越突出的作用。据农业部统计,目前全国有规模以上农产品批发市场 6000 家,其中,蔬菜和果品等鲜活农产品批发市场最多,占到批发市场总数的 52%,其次是粮食和油料市场占 16%,水产品和畜产品市场占 15%。目前,在全部农产品中蔬菜与果品经过批发市场的比例已经超过 60%,在山东等主产地更是超过了 80%。全国农产品批发市场的交易额 3461 亿元,基本构成了一个覆盖城乡,连接产地和销区的农产品流通核心网络体系。

在各类型的农产品批发市场中,国有制农产品批发市场只占全国农产品批发市场总数的 33%,集体、股份(以集体合股为主)、私有的农产品批发市场分别占全国农产品批发市场的 21%、23%、12.14%,其余所有制不明。集体、股份和私有的农产品批发市场占全国农产品批发市场总数的 57.14%。

根据商务部《中国农产品进出口月度统计报告》显示,在排除非正常因素干扰下,我国每月的农产品出口约在 18 亿至 28 亿美元间波动,如图 17 所示。而同期,农产品进口额略多于出口额,并在 15 亿至 32 亿美元间波动。

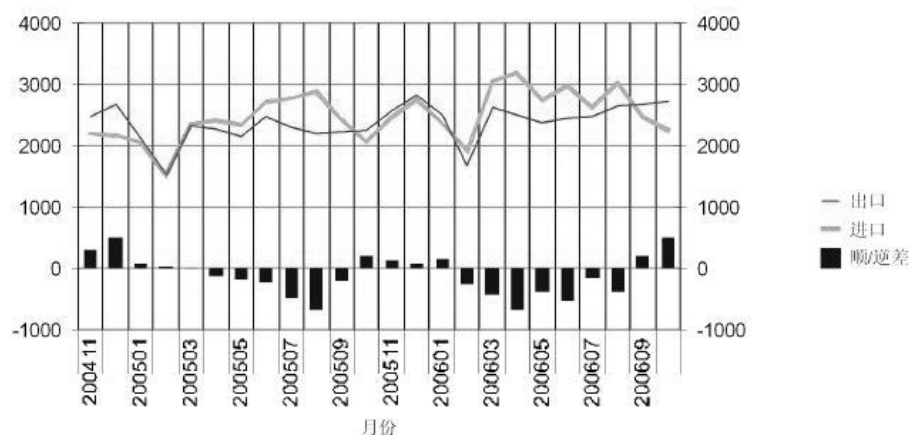


图 17 2004-2006 年农产品进出口额统计

(资料来源: 商务部《中国农产品进出口月度统计报告》)

目前,国内农产品批发市场发展迅猛,无论国有还是私营的农产品批发市场都普遍重视自身信息化建设,相继投入资金和人力开发了“数据库+网站”模式的农产品价格信息发布平台。但是,由于各个批发市场的信息发布平台基本处于各自为战的状态,因此从全国范围来看农产品批发市场尚未形成统一的信息发布平台和机制。同时,由于各批发市场在技术和人员上的限制,使其难以提供其它语言形式的价格信息服务,无法兼顾农产品国际贸易,错过大好商机。所以,有必要采取相应技术手段屏蔽各个批发市场信息发布网站在计算平台、数据存储和数据交互方面的差异,构建有效的农产品价格信息集成系统,并利用相关多语言转换方式,帮助我国农产品批发

市场尽快走出国门，适应和融入 WTO。

4.1.1 目的

针对目前农产品进出口贸易额较大以及国内农产品价格信息管理混乱等问题，本系统研发目的是利用 OGSA-DAI 数据网格技术，采用农业科技数据库集成（AST-DB）设计架构，实现在分布式异构数据环境中对农产品价格信息的集成以及对农产品价格信息的多语言转换功能。为农业科技工作者、农业主管部门、农业企业以及农民提供较完整的农产品价格信息，并进一步提高多语言的农产品价格信息服务，为促进农产品国际贸易提供技术服务。

4.1.2 应用需求分析

(1) FAO 叙词表

联合国粮农组织（Food and Agriculture Organization，FAO）提供利用 Web Services 技术的基于本体检索的叙词表。该叙词表是多语言、分结构、有控制的词汇表。它包括农、林、渔、粮食和有关领域（如环境）所有专业术语。它可以提供用户检索关键词的相关语种以及联想匹配信息。例如，我们输入“胡萝卜”，如图 18 所示，首先看到的是网站根据用户数据词汇的语种，检索出的相匹配信息；然后点击第一栏“胡萝卜”，得到所有语种关于此关键词的信息。

事实表明，FAO 的确为推动农业的信息化、现代化发展起到巨大的作用。但如果能够按照语种划分各个国家，将检索的关键词信息直接连接到各相关语言国家的农产品信息数据库中，用户享用的将绝不仅仅只是数据那么简单。

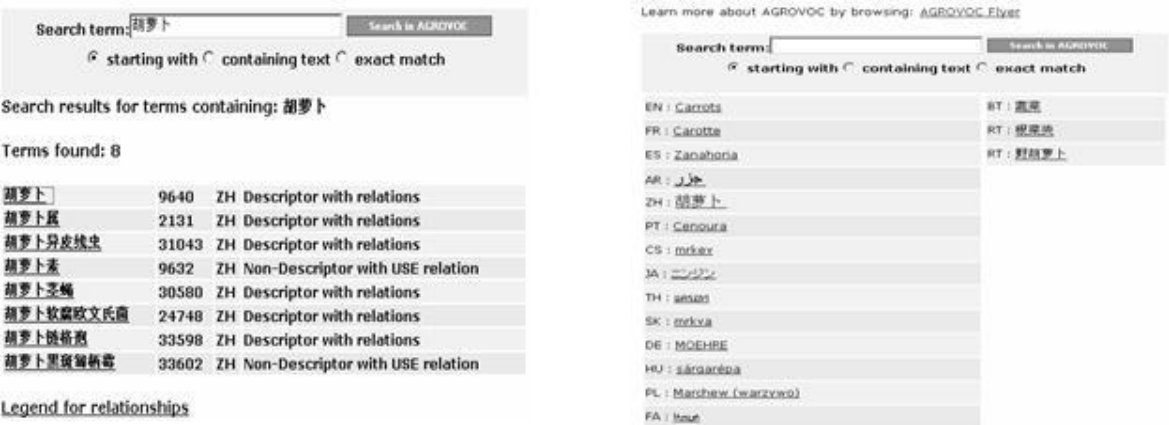


图 18 AGROVOC 截图

(2) 农业部“菜篮子”工程价格信息服务

农业部“菜篮子”工程包含 455 种农产品品种，以增加信息的透明度、提高农产品市场信息的对称性、服务老百姓为己任。从农业部的中国农业信息网上，我们可以得到每天时时更新的农产品市场信息。既可以通过各批发市场来检索，也可以通过农产品种类检索。

中国农业信息网为全国农产品的市场信息工作起到了举足轻重的作用。但是，我们也不难发

现，网站上所提供的农产品信息过于零散，缺乏良好的整合与集成，不便从数据中提取有效的市场信息，更不便于通过对农产品市场信息的掌握，对农产品价格的变更、发展趋势做出良好的推断。如图 19 所示，为中国农业信息网上 2007 年 5 月 15 日显示的胡萝卜的价格一览。

农业部-全国主要菜篮子产品批发价格日报价格列表——蔬菜类< 日期：20070603 >前5页/ [6] [7] [8] [9] [10] /后5页

品种	批发市场	单位	最低价	最高价	大家价	产地	信息员
洋白菜	武汉皇经堂	元/公斤	.4	1.3	.85		徐莉
洋白菜	邵阳江北	元/公斤	0	0	1.3		谢军
洋白菜	贵阳五里冲	元/公斤	.6	.6	.6	四川、贵州	王恒
胡萝卜	北京岳各庄	元/公斤	.6	1.4	.7		赵凤玲
胡萝卜	北京八里桥	元/公斤	.5	1	.75		王容美
胡萝卜	北京昌平	元/公斤	.4	.6	.5		师清才
胡萝卜	富怀来京西	元/公斤	.7	0	.7		孙艳霞
胡萝卜	晋太原桥西	元/公斤	.7	.8	.75		毛乐
胡萝卜	马鞍山安民	元/公斤	1	1.4	1.2		孙景秀
胡萝卜	赣九江浔阳	元/公斤	0	1.4	1.3		殷锦娥
胡萝卜	青岛城阳	元/公斤	0	0	.8		李德云
胡萝卜	郑州农产品	元/公斤	0	0	.2		赵黎伟
胡萝卜	武汉皇经堂	元/公斤	1.6	1.6	1.6		徐莉
胡萝卜	贵阳五里冲	元/公斤	1.2	1.2	1.2	福建、贵州	王恒
胡萝卜	咸阳新阳光	元/公斤	1.7	2	1.85		李博
胡萝卜	新疆乌尔禾	元/公斤	1.3	1.5	1.4		刘学刚
胡萝卜	新疆春光明	元/公斤	1.5	1.8	1.6		马文莲
木耳菜	北京岳各庄	元/公斤	2	2.4	2		赵凤玲
木耳菜	北京八里桥	元/公斤	1	1.2	1.1		王容美
木耳菜	赣九江浔阳	元/公斤	0	1.6	1.5		殷锦娥

图 19 全国主要菜篮子产品批发价格日报价格列表

（3）山东寿光蔬菜基地

“买全国、卖全国”、“没有买不到的菜，没有卖不出去的菜”是各级领导和社会各界人士对寿光蔬菜记得的评价。山东寿光蔬菜基地占地面积 600 亩，年成交蔬菜 15 亿公斤，交易额 28 亿元，年上市蔬菜品种 300 多个，全国 20 多个省、市、自治区的蔬菜来此大量交易，是中国北方最大的蔬菜集散中心、价格形成中心和信息交流中心。在国家开通的四条主要蔬菜运输绿色通道中，其中“寿光——北京”、“寿光——哈尔滨”，两条绿色通道的源头就是寿光蔬菜批发市场。

4.1.3 主要业务流程

如图 20 所示，将农业部“菜篮子”工程中价格数据库与联合国粮农组织农产品叙词表以及山东寿光蔬菜基地农产品价格数据在数据网格环境下集成的框架结构。此三者数据库将分别由代理数据库 Marketing、AGROVOC 和 Shandong 表示。科研工作人员通过 SQL 查询并集成数据库中相关信息，具体步骤如下：

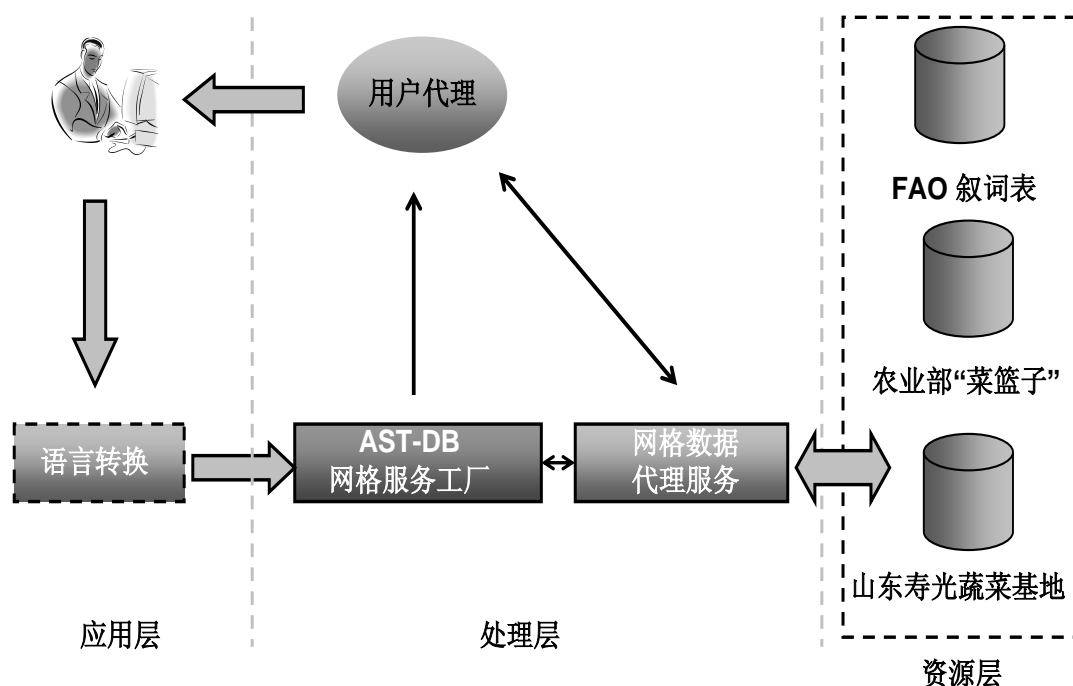


图 20 FOA_AGROVOC、MOA_Marketing 以及山东寿光蔬菜基地数据库集成示意图

- 1、用户发送农产品价格查询请求到农产品价格数据库集成系统；
- 2、系统根据用户发送请求的语言种类，将请求转换为网格环境中数据库包含的相应语种；
- 3、用户请求被转换之后，发送到 AST-DB 网格服务工厂；
- 4、AST-DB 网格服务工厂判断是否存在相应的网格数据代理服务，并创建用户代理；
- 5、AST-DB 网格服务工厂确认存在可供服务的网格服务，将网格服务句柄发送到用户代理，用户代理根据网格服务句柄请求网格服务；
- 6、得到查询结果并返回给用户。

4.1.4 主要业务用例

（1）农产品价格信息在 AST-DB 中发布用例

在此用例中，主要业务参与者为农产品价格信息资源的所有者，主要完成 Web Services 方法发布、网格服务注册等操作。

Web Services 方法发布：根据农产品价格信息的 XML 架构形式，在本数据库端生成农产品价格信息 Web Services 方法调用，具体包括某农产品批发市场的农产品品种汇总信息、某个农产品价格详细信息，以及该农产品批发市场系统用户登录和注销方法。此外，也可将相关农产品价格信息直接从数据库端转换为 XML 文件，并通过 AST-DB 网格容器直接发布数据文件。

网格服务注册：使用 AST-DB 网格容器的 ServiceGroupRegistration::ADD()方法，以 Web Services 方法 WSDL 文件向网格服务工厂注册。所要注册的服务工厂可以在该农产品价格信息资源端，也可以向其它信息资源端的网格服务工厂注册，所有注册操作均采用 DAISGR (DAIServiceGroupRegistry)。在本系统中，所有网格服务向一个中心网格服务工厂注册，该服务

工厂单独部署在一台计算机中。

(2) 价格信息多语言转换用例

在此用例中，主要业务参与者为农产品价格信息服务用户，主要完成 AGROVOC 数据库部署、Web Services 方法发布、网格服务注册等操作。

AGROVOC 数据库部署：将 FAO AGROVOC 所提供的 MySQL 数据库部署在 AST-DB 网格容器中，将 MySQL 所提供 JDBC 接口连接在服务容器的服务代理中。通过查询 AGROVOC 中关于农产品种类分类与概念间的关系（BT、NT、RT 关系），初步建立以术语码为标识的农产品种类 XML 架构，并以术语码为索引查询术语的多语言转换。

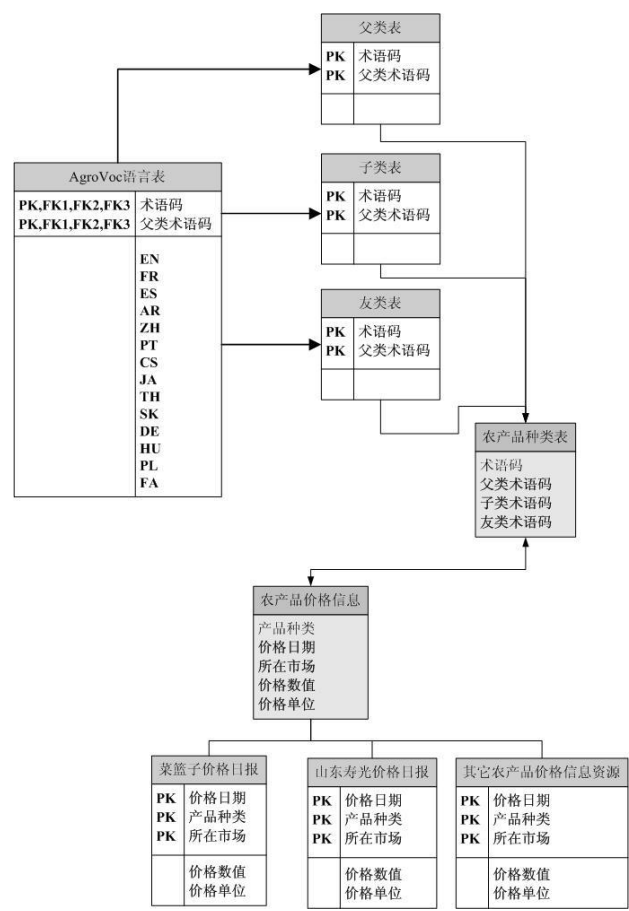
Web Services 方法发布：根据 AGROVOC 的 XML 架构形式，在本数据库端生成农产品分类和语言信息 Web Services 方法调用，具体包括某农产品品种汇总信息、某农产品多语言详细信息，以及系统用户登录和注销方法。在多语言转换服务中，利用 AGROVOC 数据库中自配的语言转换函数实现对其它语种术语对英语的转换。

网格服务注册：使用 AST-DB 网格容器的 ServiceGroupRegistration::ADD()方法，以 Web Services 方法 WSDL 文件向网格服务工厂注册。在本系统中，所有网格服务向一个中心网格服务工厂注册，该服务工厂单独部署在一台计算机中。

4.1.5 数据服务设计

(1) 实体关系

在本系统中，数据服务主要包括农产品价格信息服务、农产品分类信息服务和农产品多语言转换服务，其实体关系如图 21 所示。农产品价格信息实体的设计参照美国农业部经济研究局提供的农产品价格描述范式，并结合国内农产品批发市场价格信息发布的实际情况综合制定，以产品种类编码、市场编码和时间为标识，并以农产品价格数值和价格单位为核心属性值。各个农产品批发市场价格发布系统均参照农产品价格信息的标准格式，提供基于 XML 和 Web Services 的数据服务。AGROVOC 多语言转换实体是以术语码为标识，实现十四种语言的术语转换。农产品种类实体在父类表、友类表和子类表共同组成类似树状结构的农产品分类信息。



(2) 实体与 XML 架构对应关系

对实体关系图的描述与 XML 架构建立相应的对应关系。如图 22，即为实体与 XML 架构对应关系图。

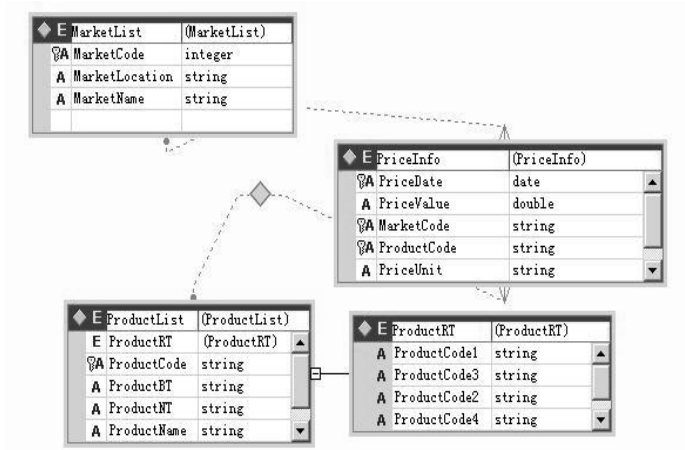


图 22 实体与 XML 架构对应关系图

(3) 农产品价格数据服务的 XML 架构描述

部分农产品价格数据 XML 架构描述，如下所示。

```

<xs:element name="MarketList" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
    </xs:sequence>
    <xs:attribute name="MarketCode" type="xs:integer" />
    <xs:attribute name="MarketLocation" type="xs:string" />
    <xs:attribute name="MarketName" type="xs:string" />
  </xs:complexType>
  <xs:key name="MarketCode">
    <xs:selector xpath="." />
    <xs:field xpath="@MarketCode" />
  </xs:key>
</xs:element><xs:element name="ProductList" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="ProductRT">
        <xs:complexType>
          <xs:sequence>
          </xs:sequence>
          <xs:attribute type="xs:string" name="ProductCode1" />
          <xs:attribute name="ProductCode3" type="xs:string" />
          <xs:attribute name="ProductCode2" type="xs:string" />
          <xs:attribute name="ProductCode4" type="xs:string" />
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="ProductCode" type="xs:string" />
    <xs:attribute name="ProductBT" type="xs:string" />
    <xs:attribute name="ProductNT" type="xs:string" />
    <xs:attribute name="ProductName" type="xs:string" />
  </xs:complexType>
  <xs:key name="ProductCodeKey">
    <xs:selector xpath="." />
    <xs:field xpath="@ProductCode" />
  </xs:key>
</xs:element><xs:element name="PriceInfo" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:complexType>
    <xs:sequence>
    </xs:sequence>
    <xs:attribute name="PriceDate" type="xs:date" />
    <xs:attribute name="PriceValue" type="xs:double" />
    <xs:attribute name="MarketCode" type="xs:string" />
    <xs:attribute name="ProductCode" type="xs:string" />
    <xs:attribute name="PriceUnit" type="xs:string" />
  </xs:complexType>
  <xs:key name="PriceInfoKey1">
    <xs:selector xpath="." />
    <xs:field xpath="@PriceDate" />
  </xs:key>
  <xs:key name="KeyPriceCode">
    <xs:selector xpath="." />
    <xs:field xpath="@ProductCode" />
  </xs:key>
  <xs:keyref name="ProductListPriceInfo" refer="ProductCodeKey">
    <xs:selector xpath="." />
    <xs:field xpath="@ProductCode" />
  </xs:keyref>
  <xs:key name="MarketCodeKey">
    <xs:selector xpath="." />
    <xs:field xpath="@MarketCode" />
  </xs:key>
  <xs:keyref name="MarketListPriceInfo" refer="MarketCode">
    <xs:selector xpath="." />
    <xs:field xpath="@MarketCode" />
  </xs:keyref>

```

</xs:element>

4.2 农产品价格数据集成 AST-DB 访问管理

基于数据网格技术的农产品价格数据集成 AST-DB 与基于分布式数据库查询技术之间的差异,除了数据访问方面的功能区别,在数据访问安全方面也有差异。主要体现在访问用户的权限设置问题。在网格环境中访问数据与在普通环境下访问数据有较多差异,其中之一,就是每个网格数据服务节点用户向数据库中添加数据的权限,但是此权限的使用不可能面向所有网格数据用户。为此,在本篇论文农产品价格集成 AST-DB 中根据农产品价格系统的自身特点,将用户登录权限设置成以下两大类别:

4.2.1 部分网格数据管理权限的用户访问管理

在农产品价格数据集成的整体框架结构中,除了有相应主管部分负责当地农产品价格数据的管理职能以外,该农产品价格数据应该是面向所有农产品价格数据的使用者。在此过程中就涉及到用户登录权限的问题。本篇论文中,我们将面向只拥有部分或不拥有网格数据管理权限的用户登录成为 GridDB 登录。

根据农产品价格管理职能部门的要求,可以将登录 GridDB 的用户分为如下两类:

(1) 拥有地方农产品价格网格数据服务的管理功能。对于此类管理部门,我们会按照该管理部门的要求,授予相应数据库数据删除、添加、更改功能。该授权往往只能面向本地农产品价格数据库。

(2) 只拥有农产品价格网格数据服务的访问功能。以农户的角度来说,单一访问农产品价格已经能够满足农户对农产品价格的需求。

此外,在此向登录权限设置中,也可以按照对个例数据库的权限设置,比如农产品价格管理部门,我们可以只授予农业部“菜篮子”工程数据库访问权限。

4.2.2 全局网格数据管理权限的用户访问管理

农产品价格数据集成 AST-DB 登录中设置 DBMS 用户登录。拥有全部网格数据管理权限的登录用户可以享用此项功能,拥有从国内外到各省市地方的农产品价格网格数据的管理权限,统筹、监督全局农产品价格数据。

4.3 农产品分类与多语言转换 AST-DB 网格服务数据访问

从农产品市场数据源到访问用户的,如果忽略中间环节以及在各环节中的功能和流程的话,图 23 以形象具体的方式描述出农民用户与农产品价格数据源之间的关系。

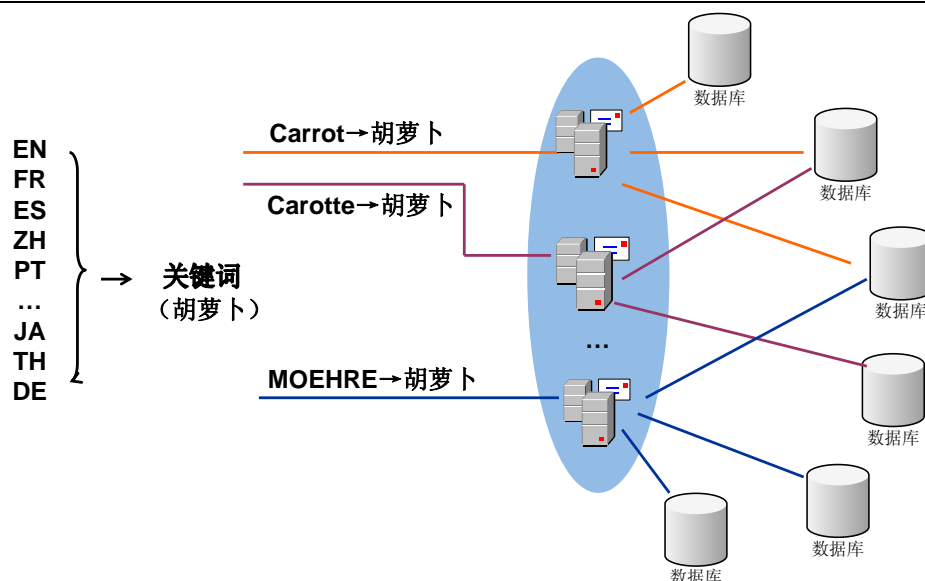


图 23 农产品价格概要访问

从以往访问多个网址，人工合成农产品市场信息，到访问本地网格数据服务，一览农产品价格市场全局，数据网格技术的发展起到举足轻重的作用。它有效解决了跨地域、跨数据库问题，屏蔽数据类型异构性与分布性特征，提取便捷、快速的信息查询。同时，需要说明的是，基于数据网格技术的农产品价格数据集成与分布式数据库查询的不同体现在：分布式数据库查询是通过关键词到分布在各地的数据库之间进行查询检索，该检索无法屏蔽数据格式异构问题，并且在数据体现上也存在延迟；基于数据网格技术的农产品价格数据集成则可以通过对各注册数据库的服务发现功能，实现数据实时更新，及时准确地反映农产品市场价格问题。在网格环境下，对用户农产品价格访问大致划分为以下三类。

4.3.1 单数据库网格数据服务访问

将农产品价格信息数据导入数据库，这里需要说明的是，农业科技数据库网格数据服务的构建是基于 OGSA-DAI、Web Services 服务器（Axis 2.0）以及 Globus Toolkit 4.0 基础之上。在成功配置 Globus Toolkit 4.0 并将其部署到 tomcat 之后，安装 OGSA-DAI。从理论上来说，OGSA-DAI 本身可以提供基础的数据查询与集成。单纯基于 OGSA-DAI 的农业科技数据库集成系统，允许通过开源软件的安装部署屏蔽底层各种数据资源的差别。但是，DAI 本身仅实现了数据库的 Web Services 访问模式，具体针对不同的数据库还需要编写不同的接口代码⁵。本论文采用 MySQL 类型数据库为数据源。

部署 OGSA-DAI 之后，正确输入命令行，我们能够得到“数据服务浏览器”的 GUI 界面。如图 24 所示，以农产品价格信息为例，在“查询语句”中输入“Select * from AGROVOC where Agricultural ProductId = 541”或者“Select * from AGROVOC where Agricultural ProductName = ‘胡萝卜’”我们就可以得到 ProductId 为 541 的胡萝卜价格信息。

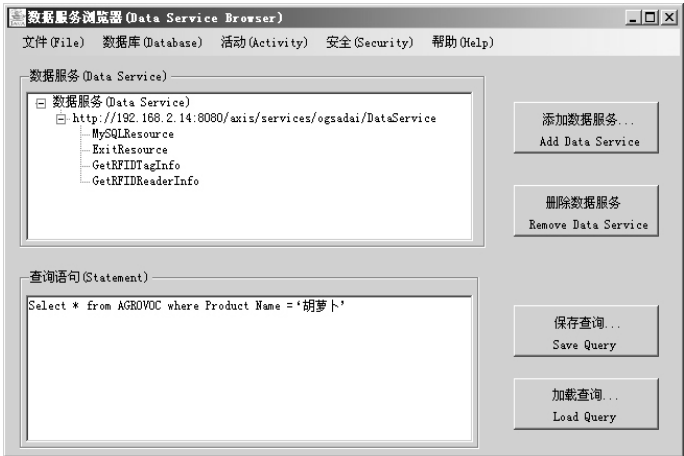


图 24 单数据库网格数据服务访问

在数据网格环境下，数据服务的注册与发现遵循动态机制。科技人员在查询农产品价格信息时，并不知道自己所需要的服务是由哪些数据服务结点提供，更不知道数据服务结点的位置，所以网格数据服务还提供注册“添加数据服务”和“删除数据服务”功能，保持数据服务实时更新。通过输入数据服务 URL 地址，“添加”或“删除”数据服务。

4.3.2 双数据库网格数据服务访问

部署 OGSA-DAI、Web Services 服务器（Axis 2.0）以及 Globus Toolkit 4.0 基础之后，正确输入命令行，我们能够得到“数据服务浏览器”的 GUI 界面。“数据服务浏览器”提供“联合查询”服务。联合查询主要实现双数据库网格数据服务访问，它可以将分布数据库中得到的查询结果汇集到目标数据库中，保持数据更新，减少数据冗余。如图 25 所示，在数据库 A 中查询 ProductId = 541 的农产品，在数据库 B 中查询“胡萝卜”，建立 ProductId = 541 AND ProductName = “胡萝卜”的关系，可以将双方数据库为相应数据库中数据建立匹配关系。需要说明的是，此关系建立的前提是在双方数据库中数据格式相同的前提条件下。

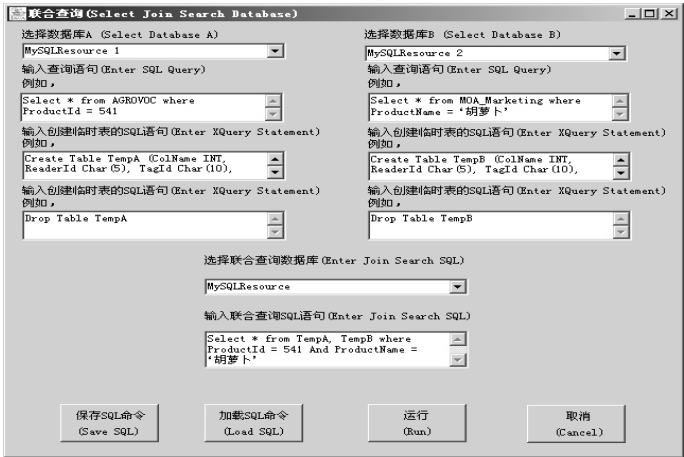


图 25 双数据库网格数据服务访问

4.3.3 多数据库网格数据服务访问

多数据库网格数据服务访问形式与单数据库以及双数据库访问形式存在很大区别。区别之一在于进行多数据库访问集成的时候遇到的访问标准问题。因为全球各地农产品价格数据的数据格式不统一，访问数据的格式和关键词也会因为语言不同存在差异，怎样屏蔽数据格式差异带来的问题以及如何解决由于语言不同造成访问数据库受限制？FAO 叙词表提供了一个很好的解决方案——利用农产品价格元数据将农产品价格表示形式统一。

AGROVOC 对农产品分类已经形成标准化，主要是按照农产品编码以及农产品名称为关键词进行访问。并且在 AGROVOC 中查询词条，AGROVOC 会按照本身已涉及的语言种类，提供有关该词条的多语种查询结果，并且依据该词条在 AGROVOC 所属分类等级，提供与该词条相关的其他信息。AGROVOC 对词条信息分类方式如下：BT(board term)、NT(narrower term)、RT(related term)、UF(non-descriptor)。

农产品分类与多语言转换 AST-DB 网格数据服务访问中，按照 AGROVOC 提供的解决方案，以农产品名称为关键词查询。当在搜索框中输入“胡萝卜”搜索之后，AGROVOC 会按照该关键词提供叙词表中涉及信息将“胡萝卜”转换为各类语言关键词。这样，通过点击不同语种的关键词——“胡萝卜”，我们便可以访问不同语种的农产品价格数据。



图 26 多数据库访问——以汉语作为检索词语种



图 27 多数据库访问—以英语作为检索词语种



图 28 多数据库访问—以法语作为检索词语种

4.4 农产品价格 AST-DB 网格服务意义

利用数据网格技术解决现存农产品市场的价格信息不对称问题，并且有针对性地选择将 FAO 叙词表与农业部“菜篮子”工程结合，目前，此例实现对于农产品市场有一定程度上的意义，如

下：

第一，从数据到信息再到知识的过渡过程中，数据集成对最终知识的获取至关重要。尤其，在将利用 Web Services 技术的 FAO 叙词表与农业部“菜篮子”相结合的设想中，能够使 FAO 通过关键字获得全球某处（如中国）农产品价格数据。农产品价格数据是在分析当地农业状态的关键信息，通过农产品价格数据的波动，可以推测数据来源地的气候变化、自然灾害、土壤变化、肥料因子、病虫害以及农产品供需变化等。

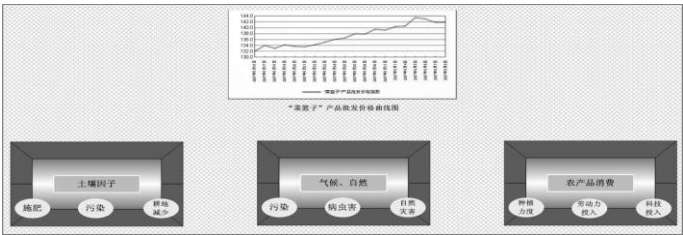


图 29 农产品价格关联因素示意图

第二，农业科技数据库集成的意义（以此 AST-DB 农产品价格集成为例），与同类处理分布式数据资源的论述相比较，不但提供了分布式数据检索功能，更在检索数据的基础上，将有效数据整合利用；

第三，从被动查询到主动获取，从分散查询到有效的整体访问，AST-DB 农产品价格集成从某种程度上改变了数据库访问方式；

第四，基于 Web Services 技术和数据网格技术，实现农产品价格的多语言查询，整合农产品价格信息资源，促进国家间的贸易合作与协调；

第五，通过该集成事例，说明本文所论述的农业科技数据与农业科学数据库之间的细微区别，即科学数据侧重科学基础数据，科技数据侧重实际应用数据。

第五章 总结与展望

数据网格技术，作为一项目前全球范围内开展的技术研究领域，为推动新领域的研究拓展作出巨大的贡献。这首先归功于网格技术是一项服务于全人类的开源技术。这为本论文作者顺利开展农业领域的网格技术研究起到了举足轻重的推动作用。

5.1 工作总结

基于 OGSA 的农业科技数据库集成是网格技术应用于农业领域的一种尝试。本文从基本概念着手，学习网格技术领域中的各有关概念。了解关于网格技术的应用领域分类、技术分类等，对数据网格的架构也是由浅入深，从五层沙漏架构到后来的开放式网格服务体系结构，再到本论文用到的主要技术——OGSA-DAI。这些尝试，无不为本论文的成文奠定了基础，也为论文的论述展开做好了准备工作。

论文以分析农业科技数据现存的问题开篇，分析农业科技数据的自身特点，简要介绍目前关于数据集成的普遍方法。同时，从实际出发，评估农业科技数据在具体集成实施中的问题。论文着重从 Globus Toolkit 和 OGSA 等技术方面入手，分析开源网格基础平台 Globus 的更新发展，并从中作出适用于论文研究内容的基础平台选择，简要介绍了 OGSA、OGSA-DAI 以及 Web Services。关于农业科技数据库集成设计方面，本篇论文根据数据库集成可能面临的问题，提出解决方案、作业流程设计、编程设计、功能设计、服务交互设计以及应用架构设计。在选取理论与实践结合点时，本文以农产品价格数据集成为例，利用数据集成理论将 FAO 叙词表与中华人民共和国农业部“菜篮子”工程以及山东寿光蔬菜基地农产品价格数据结合，初步设计用户登录管理和数据库集成访问两类服务。在用户登录管理中，本文针对用户特征将其分为部分用户登录管理和全局用户登录管理，以区分用户权限问题。在数据库集成访问服务中，本文也对数据库集成作了大致分类，单数据库服务访问、双数据服务访问以及多数据服务访问。

5.2 下一步工作

在网格技术领域研究工作初步告一段落之时，论文存在许多需要进一步完成的工作任务：

(1) 本文实现农产品价格数据集成 AST-DB 用户管理和网格服务数据访问两类服务，但对网格服务数据添加、编辑以及如何通过用户权限变更解决用户访问不一致问题有待继续研究；

(2) 本文中农产品价格数据集成是基于 MySQL 数据库类型，作者建议下一步研究实现通过 URL，以访问 Web 数据库的形式集成，用以解决建立固定类型数据库问题；

(3) 在 FAO 叙词表与农业部“菜篮子”相结合的实例仍需继续探索，除了实现分布式异构数据集成访问，应详细深入探讨多语言转换涉及的 Web Services 技术；

(4) 在数据库集成过程中，涉及到异构数据库资源描述问题以及数据安全利用和数据资源有效调度问题，这些问题的存在必然会影响网格技术在实际中的应用。

附 录

- Globus Toolkit 4 部署所需软件包:

JDK (J2SE1.5), jdk-1_5_0_06-windows-i586-p.exe

Jakarta-ant, apache-ant-1.7.0-bin.zip

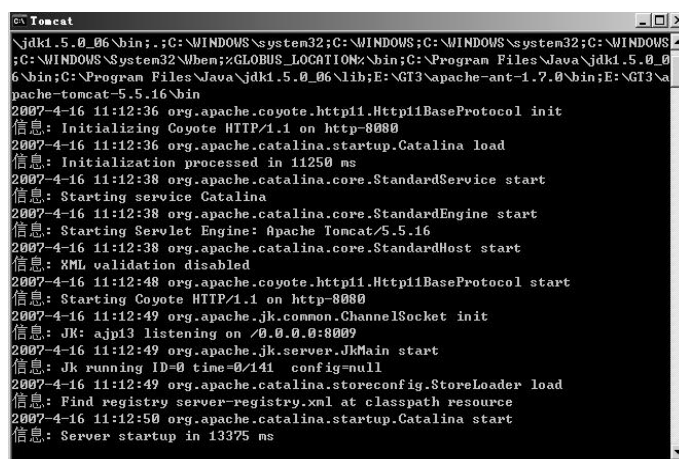
Tomcat, apache-tomcat-5.5.16.zip

Axis, axis-bin-1_3.zip

GT 4.0.2, 在 windows 平台下安装要选择 ws-core-4.0.2

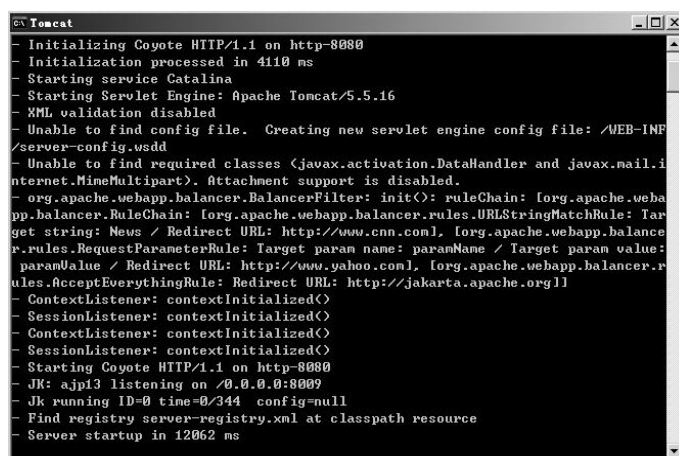
- Globus Toolkit 4 安装、配置

安 装 jdk-1_5_0_06-windows-i586-p.exe , 配 置 apache-ant-1.7.0-bin.zip 和 apache-tomcat-5.5.16.zip。命令行, 在 tomcat 目录 bin\下执行 startup.bat 启动;



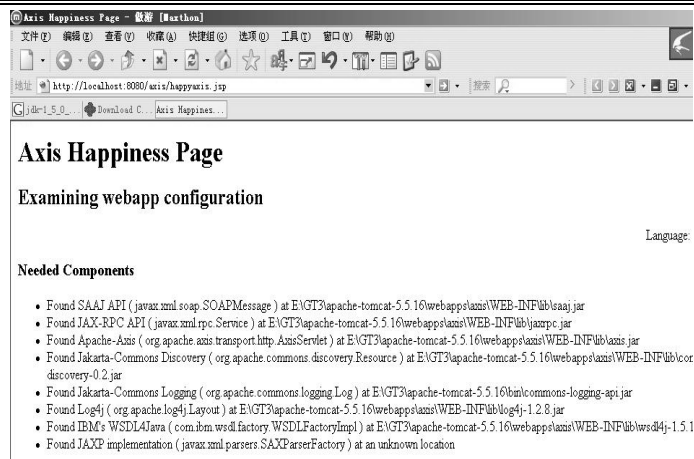
```
Tomcat
C:\jdk1.5.0_06\bin;.C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\system32;C:\WINDOWS\
C:\WINDOWS\system32\cmd;GLOBUS_LOCATION\bin;C:\Program Files\Java\jdk1.5.0_0
6\bin;C:\Program Files\Java\jdk1.5.0_06\lib;E:\GT3\apache-ant-1.7.0\bin;E:\GT3\ap
ache-tomcat-5.5.16\bin
2007-4-16 11:12:36 org.apache.coyote.http11.Http11BaseProtocol init
信息: Initializing Coyote HTTP/1.1 on http-8080
2007-4-16 11:12:36 org.apache.catalina.startup.Catalina load
信息: Initialization processed in 11250 ms
2007-4-16 11:12:38 org.apache.catalina.core.StandardService start
信息: Starting service Catalina
2007-4-16 11:12:38 org.apache.catalina.core.StandardEngine start
信息: Starting Servlet Engine: Apache Tomcat/5.5.16
2007-4-16 11:12:38 org.apache.catalina.core.StandardHost start
信息: XML validation disabled
2007-4-16 11:12:48 org.apache.coyote.http11.Http11BaseProtocol start
信息: Starting Coyote HTTP/1.1 on http-8080
2007-4-16 11:12:49 org.apache.jk.common.ChannelSocket init
信息: JK: ajp13 listening on /0.0.0.0:8009
2007-4-16 11:12:49 org.apache.jk.server.JkMain start
信息: Jk running ID=0 time=0/141 config=null
2007-4-16 11:12:49 org.apache.catalina.storeconfig.StoreLoader load
信息: Find registry server-registry.xml at classpath resource
2007-4-16 11:12:50 org.apache.catalina.startup.Catalina start
信息: Server startup in 13375 ms
```

测试: 在浏览器输入 <http://localhost:8080>; 如果成功将显示 Tomcat 欢迎页。安装配置 axis-bin-1_3.zip, 重新启动 tomcat;

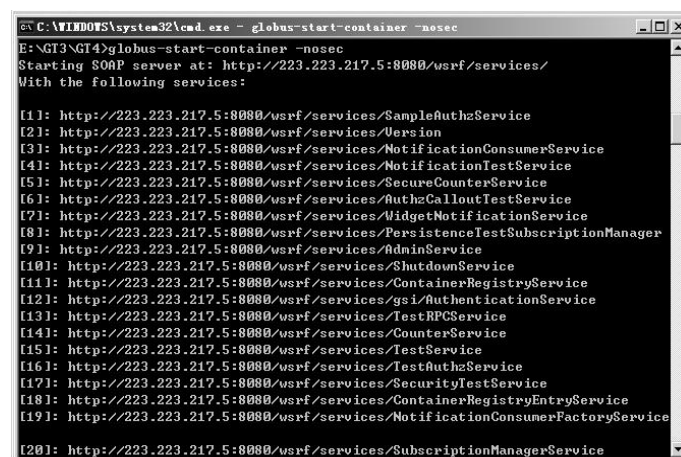


```
Tomcat
- Initializing Coyote HTTP/1.1 on http-8080
- Initialization processed in 4110 ms
- Starting service Catalina
- Starting Servlet Engine: Apache Tomcat/5.5.16
- XML validation disabled
- Unable to find config file. Creating new servlet engine config file: /WEB-INF
/server-config.wsdd
- Unable to find required classes (javax.activation.DataHandler and javax.mail.i
nternet.MimeMultipart). Attachment support is disabled.
- org.apache.wehapp.balancer.BalancerFilter: init(): ruleChain: [org.apache.wehapp.balancer.RuleChain: [org.apache.wehapp.balancer.rules.URLStringMatchRule: Target string: News / Redirect URL: http://www.cnn.com], [org.apache.wehapp.balancer.rules.RequestParameterRule: Target param name: paramName / Target param value: paramValue / Redirect URL: http://www.yahoo.com], [org.apache.wehapp.balancer.rules.AcceptEverythingRule: Redirect URL: http://jakarta.apache.org]]
- ContextListener: contextInitialized()
- SessionListener: contextInitialized()
- ContextListener: contextInitialized()
- SessionListener: contextInitialized()
- Starting Coyote HTTP/1.1 on http-8080
- JK: ajp13 listening on /0.0.0.0:8009
- Jk running ID=0 time=0/344 config=null
- Find registry server-registry.xml at classpath resource
- Server startup in 12062 ms
```

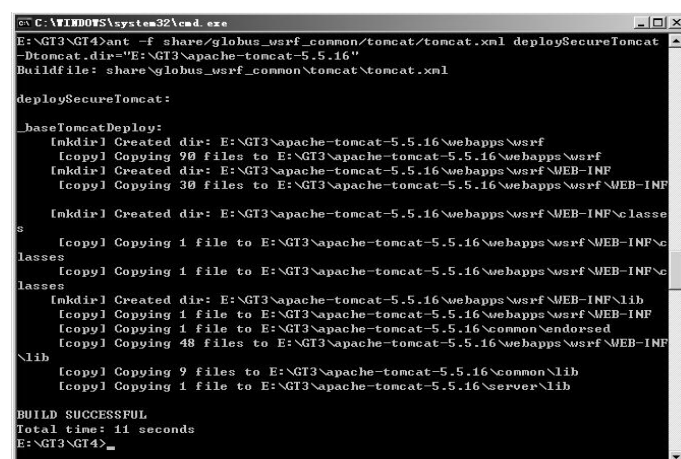
测试: 在浏览器输入 <http://localhost:8080/axis/happyaxis.jsp>; 如果能访问表示设置成功。



配置 ws-core-4.0.2-bin.zip，输入命令行，globus-start-container -nosec



部署 GT4 到 tomcat:

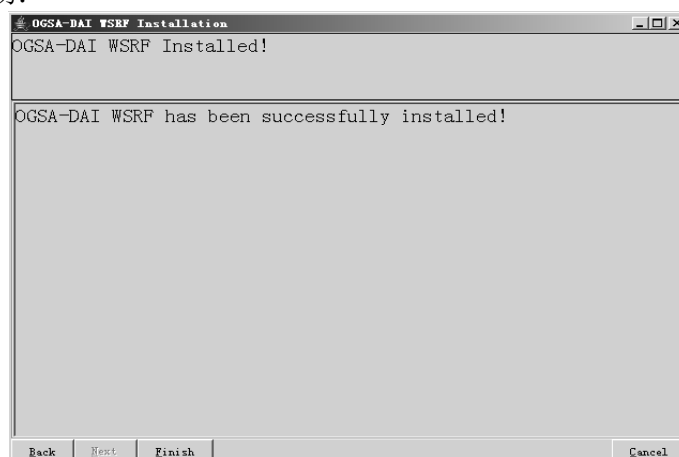


测试：在浏览器输入 http://localhost:8080/wsrf/services；如果能显示表示成功。

And now... Some Services

- NotificationTestService (*wsdl*)
 - generateNotification
 - selfSubscribe
- TestAuthService (*wsdl*)
 - addDeclineMethod
 - SAMLRequest
- TestServiceWrongWSDL (*wsdl*)
 - createResource
 - resetNumInstances
 - getInstanceInfo
 - testLocalInvocation
- AuthzCalloutTestService (*wsdl*)
 - getValue
 - setValue
 - createResource
 - setAuthz
 - noAuth
 - gssSecConnDeleg
 - gssTransportIntegrity
 - gssTransportPrivacy
 - gssTransportOnly
 - gssSecConnIntegrity
 - gssSecConnPrivacy
 - gssSecConnOnly
 - gssSecMsgOnly
 - gssSecMsgPrivacy
 - gssSecMsgIntegrity
 - gssSec

下载 ogsadai-wsrf-2.2-bin，并复制、设置，运行 tomcat 输入命令行：cd ogsa-dai ant guiInstall，OGSA-DAI 安装成功：



参考文献

1. 蔡剑, 景楠. Java Web 应用开发: J2EE 和 Tomcat(第二版) [M]. 清华大学出版社, 2005.1~12
2. 曹顺良. 生物学数据集成若干关键问题研究[D]. 复旦大学, 2004 年 11 月.
3. 常永平. 农业信息化: 冲刺“最后一公里”[ER/OL]. [http:// nc. people. com. cn/ GB/ 61160/ 5215939.html](http://nc.people.com.cn/GB/61160/5215939.html), 2005
4. 福莱斯汀, 约瑟夫, 张少华, 战晓苏. 网格计算/Grid Computing[M]. 北京: 清华大学出版社, 2005.107~110
5. 郝卫东, “十一五”高等学校应用型规划教材, 《网络环境下的电子商务与电子政务建设》[M]. 北京: 清华大学出版社, 2006, 119.
6. 何坚勇. 运筹学基础[M]. 北京: 清华大学出版社, 2000: 355—386.
7. 黄理灿, 刘元安. e-Science、网格及其可扩展性体系结构[M]. 北京: 北京邮电大学出版社. 2005.207~220.
8. 姜启源. 数学模型 (第三版) [M]. 北京: 高等教育出版社, 1992. 111~116.
9. 金海, 中国教育科研网格图像处理网格应用平台设计规范[M]. 北京: 清华大学, 2006. 98~102.
10. 李春葆, 曾慧. 数据库原理与应用[M]. 清华大学出版社, 2005. 56~68
11. 李仁庆, 李红星. 基于神经网络技术的异构数据库集成查询[J]. 大连轻工业学院学报. 22(2), 2003: 146-150.
12. 李效东. 自治异构数据源的集成查询处理[D]. 中国科学院研究生院, 2002 年 4 月
13. 李志刚. 面向数据库网格的数据集成与查询技术研究[实现][D]. 中国国防科技大学, 2005.
14. 林行健, Oracle 10g 数据库管理、应用与开发[M]. 北京: 清华大学出版社, 2005. 219~250.
15. 麦克多纳德, 戢中东. .NET 分布式应用程序: 集成 XML Web 服务与 .NET 远程处理/微软.NET 程序员系列/Microsoft.NET Distributed [M]. 北京: 清华大学出版社, 2005. 48~50.
16. 孟宪学. 国家农业科学数据中心的设计与建设研究[J], 农业图书情报学刊, 2004(12).
17. 诺克斯, Oracle Database 10g 安全性高效设计[M]. 北京: 清华大学出版社, 2006. 106~109.
18. 苏新宁. 数据仓库和数据挖掘/信息分析丛书[M]. 北京: 清华大学出版社, 2006. 315~216.
19. 徐冠华. 实施科学数据共享增强国家科技竞争力[J]. 中国基础科学, 2003(1): 5-9.
20. 徐志伟, 冯百明, 袁平鹏, 福斯特, 金海, 石柯. 网格计算技术[M]. 北京: 电子信息技术专著出版专项. 2004.223~253.
21. 徐志伟, 冯百明, 李 伟. 网格计算技术[M]. 北京: 电子工业出版社, 2004. 69~73.
22. 杨利, 昌月楼. 并行数据库技术[M]. 长沙: 国防科技大学出版社, 2000. 110~115.
23. 中国互联网络信息中心. 中国互联网络发展状况统计报告[M]. 北京: 中国互联网络信息中心, 2007 年 1 月, 11~13.
24. 中国计算机学会, 2004 中国计算机科学技术发展报告[M]. 北京: 清华大学出版社, 2004.46~50.
25. 中国计算机学会, 2004 中国计算机科学技术发展报告[M]. 北京: 清华大学出版

- 社,2004.68~70.
26. 周园春. 科学数据网格分布式查询框架及其关键技术研究[D]. 中国科学院研究生院, 2006年4月.
 27. Akhil Sahai. Web Services In The Enterprise: Concepts, Standards, Solutions, and Management [M].Sven Graupner, 2005. 33~34.
 28. Apache. Web Services Resource Framework [EB/OL]. <http://ws.apache.org/wsrf/wsrf.html>, 2005
 29. Argonne Lab, Access Grid [EB/OL]. <http://www.accessgrid.org/>, May, 2007
 30. Argonne National Library, Globus Toolkit 2 [EB/OL]. [www. anl. gov/ techtransfer/ docs/ GlobusToolkt_2new_logo.doc](http://www.anl.gov/techtransfer/docs/GlobusToolkt_2new_logo.doc), 2006
 31. Borja Sotomayor. Globus Toolkit 4: Programming Java Services [M]. Morgan Kaufmann Publisher, 2006. 32~33.
 32. P. Brittenham. An Overview of the Web Services Inspection Language [EB/OL]. [http:// www. ibm. com/developerworks/webservices/library/ws-wslover](http://www.ibm.com/developerworks/webservices/library/ws-wslover), 2006
 33. Buck-ER, Galimow J.SAP R/3 System, A Client/Server Technology [M]. Addison-Wesley, Reading, MA, 1996.46~51.
 34. Cardelli L, Gordon D. Mobile Ambients, Foundations of Software Science and Computational Structure, LNCS 1378, Springer-Verlag, 1998:140-155.
 35. Christensen, E., Curbera, F., Meredith, G. and Weerawarana, S. (2001). Web Services Description Language (WSDL)1.1[EB/OL].W2C Note 15, <http://www.w3.org/TR/wsdl>, 2005
 36. Czajkowski, K., Ferguson, D.F., Foster, I., Frey, J., Graham, S., Sedukhin, I., Snelling, D., Tuecke, S. and Vambenepe, W. (March 2004). The WS-Resource Framework, Version 1.0, <http://www-106.ibm.com/developerworks/library/ws-resource/ws-wsrf.pdf>.
 37. De Roure, D., Baker, M.A., Jennings, N. and Shadbolt, N.The Evolution of the Grid, in Grid Computing: Making the Global Infrastructure a Reality [M]. England: Wiley Series in Communications Networking and Distributed Systems, April 2003. 65~100.
 38. Dialani, V., Miles, S., Moreau, L., Roure, D.D. and Luck, M. Transparent Fault Tolerance for Web Services Based Architectures[R]. Proceeding of 8th International Europe Conference, Paderborn, Germany. Lecture Notes in Computer Science, Springer-Verlag, 2003.
 39. Foster, C. Kesselman. The Physiology of the Grid: An Open Grid Services Architecture for Distributed System Integration. Globus Project. January, 2002. 30~32
 40. Foster, I., Frey, J., Graham, S., Tuecke, S., Czajkowski, K., Ferguson, D., Leymann, F., Nally, M., Sedukhin, I., Snelling, D., Storey, T., Vambenepe, W. and S.Weerawaran, Modeling Stateful Resources with Web Service, Version 1.1. [http :// www- 106. ibm. com/ developerworks/ library/ ws-resource/ws-modelingresources.pdf](http://www-106.ibm.com/developerworks/library/ws-resource/ws-modelingresources.pdf).
 41. Foster, I., Kesselman, C. and Tuecke, S., The Anatomy of the Grid: Enabling Scalable Virtual Organizations[D], International Journal of Supercomputer Applications, 15(3), 2001
 42. Foster, I., Kesseslman, C., Nick, J. and Tuecke, S. (June 2002). The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration, [http:// www. globus. org/](http://www.globus.org/)

- research/papers/ogsa.pdf.
43. Fran Berman of UC San Diego (UCSD) and the San Diego Supercomputer Center (SDSC), National Partnership for Advanced Computational Infrastructure [EB/OL]. [http:// www. npaci. edu/ About_NPACI/](http://www.npaci.edu/About_NPACI/), May, 2007
 44. Franklin M, Zdonik S. Data in your face: Push technology in perspective[C]. In Proceedings of the ACM SIGMOD Conference on Management of Data. Seattle, Wash, June 1998, 516–519.
 45. Frey J., Graham S., Czajkowski K, Foster I., Leymann F., Maguire T., N. Nally, M. Storey, T. Sedukhin, I. Snelling, D. Tuecke, S. Vambenepe, W. and S.Weerawarena, [EB/OL]. Web Services Resource Lifetime, Version 1.1. [http:// www- 106. ibm. com/ developerworks/ Library/ wa-resource/weresourcelifetime.pdf](http://www-106.ibm.com/developerworks/Library/wa-resource/weresourcelifetime.pdf), 2006
 46. Funded by European Union. The [EB/OL]. www.cern.ch/eu-datagrid, April, 2007
 47. Gannon D., A Revised Analysis of the Open Grid Services Infrastructure, Journal of Computing and Informatics, 21, 2002, 321~332, [http: //www. extreme, indiana, edu/ ~aslom. papers. ogsa_ analysis4.pdf](http://www.extreme.indiana.edu/~aslom.papers.ogsa_analysis4.pdf).
 48. GGF OGSA-DAI Group. Open Grid Services Architecture – Data Access Integration [EB/OL]. <http://www.ogsadai.org.uk/about/ogsa-dai/>, 2006.
 49. Globus, Globus Toolkit 2.4 Release Manuals [EB/OL]. <http://www.globus.org/toolkit/docs/2.4/>, 2006
 50. Globus, Open Grid Service Architecture [EB/OL]. <http://www.globus.org/ogsa/>, 2005
 51. Globus, the WS-Resource Framework [EB/OL]. www.globus.org/wsrf/, 2006
 52. Globus. Globus: European Data Grid[EB/OL]. www.globus.org/alliance/news/EDG-index.html
 53. Globus. The Globus Toolkit 3 Programmer's Tutorial [EB/OL]. www.gdp.globus.org/gt3-tutorial/multiplehtml/index.html - 6k, 2006.
 54. Globus. The GT3 Tutorial [EB/OL]. <http://gdp.globus.org/gt3-tutorial/>, 2005
 55. Gopalan, S.R. A Detailed Comparison of CORBA, DCOM, and Java/RMI [K]. Object Management Group (OMG), September 1998, 12~14.
 56. Grid Checklist, What is the Grid? A Three Point Checklist [EB/OL]. [http:// www. gridtoday. com/ 02/0722/100136.html](http://www.gridtoday.com/02/0722/100136.html), 2005.
 57. Grid Infrastructure Group .Tera Grid [EB/OL]. <http://www.teragrid.org/>, May 2007.
 58. Guo Man, Zhu Haipeng. Ontology-Based Agricultural Scientific Data Integration in OGSA-DAI Environment [J]. Asian Federation for Information Technology in Agriculture, India, 2006.
 59. Heng Tao Shen. Advanced Web and Network Technologies and Applications[R] APWeb 2006 International Workshops, 2006, Harbin, China, Springer, 2006.
 60. I Foster, C Kesselman. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International J. Supercomputer Applications, 15(3), 2001. 75~79.
 61. I Foster, C Kesselman. The Grid: Blueprint for a Future Computing Infrastructure. San Francisco, USA: Morgan Kaufmann Publishers, 1999. 84~89.
 62. Ian Foster and Carl Kesselman (eds). The Grid: Blueprint for a New Computing Infrastructure,

- 1st edition [M]. San Francisco, USA: Morgan Kaufmann Publishers, November 1998. 220~236.
63. IBM Globus Computing, A visual tour of Open Grid Services Architecture [EB/OL]. <http://www.ibm.com/developerworks/grid/library/gr-visual/>, 2006
64. IBM Grid Computing, A developer's overview of OGSi and OGSi-based grid computing [EB/OL]. www.ibm.com/developerworks/library/gr-ogsi/, 2006
65. IBM Grid Computing, Grid Computing home page [EB/OL]. [http:// www- 03.ibm. com/ grid/?re=gi67fss](http://www-03.ibm.com/grid/?re=gi67fss), 2005
66. J. Smith, A. Gounaris, P. Watson, N. Paton, A. Fernandes, R. Sakellariou. Distributed Query Processing on the Grid[C]. In Proceedings of Grid Computing 2002, Springer, LNCS 2536, 2002: 279-290.
67. J.A. Senn. 信息技术基础 Information Technology: Principles, Practices, Opportunities (第3版影印版) [M]. 清华大学出版社, 2005年11月. 519~520
68. Jenq B, Woelk D, Kim W, Lee W. Query processing in distributed ORION. In Proceedings of the International Conference on Extending Database Technology (EDBT). Venice, Italy, March 1990:109-127.
69. Jenq B, Woelk D, Kim W, Lee W. Query processing in distributed ORION. In Proceedings of the International Conference on Extending Database Technology (EDBT). Venice, Italy, March 1990:169-187
70. Julia Lerman. The Data Farm [EB/OL]. [April, 2007]. <http://www.thedatafarm.com/main.aspx>.
71. Kunszt, Peter Z. (April 2002). The Open Grid Services Architecture – A Summary and Evaluation[EB/OL].[2006].<http://edms.cem.ch/file/350096/1/OGSAreview.pdf>.
72. M. Alpdemir, A. Mukherjee, N. Paton, P. Watson, A. Fernandes, A. Gounaris, J. Smith. Service-based Distributed Querying on the Grid[C]. In Proceedings of the First International Conference on Service Oriented Computing. Springer, December 2003, 467-482.
73. Maozhen Li, Mark Baker. The Grid Core Technologies [M]. 清华大学出版社, 2006
74. Michael H. Böhlen. E-Government[R] International Conference, TCGOV 2005, Bolzano, Italy, Springer, 2005.
75. Nogu, William A. and Ballinger, Keith. (November 2001). The WS-Inspection and UDDI Relationship[EB/OL].[2006].<http://www-106.ibm.com/developerworks/webser-vices/library/ws-wsiluddi.html>.
76. NASA. NASA Advanced Supercomputing (NAS) Division [EB/OL]. [www. nas. nasa. gov/ About/ IPG/ipg.html](http://www.nas.nasa.gov/About/IPG/ipg.html) , May 2007
77. OGSi, Oxygen Generator Manufacturer [EB/OL]. www.ogsi.com/, 2005.
78. Papakonstantinou Y, Garcia MH, Widom J. Object exchange across heterogeneous information sources. In Proceedings of the IEEE Conference on Data Engineering Taipei, Taiwan, 1995:251-260.
79. R Buyya, D Abramson, J Giddy. An Economy Driven Resource Management Architecture for Global Computational Power Grids[C].In:Proceedings of the 2000 International Conference on

- Parallel and Distributed Processing Techniques and Applications (PDPTA 2000), Las Vegas, USA, CSREA Press, USA, 2000-06. 86~90.
80. R Buyya, D Abramson, J Giddy. Nimrod/G: Architecture for a Resource Management and Scheduling System in a Global Computational Grid [C]. In: The 4th International Conference on High Performance Computing in Asia-Pacific Region (HPC Asia 2000), 2000, 55~58.
81. Rowe, Schuh. Computer Networking [M]. 北京: 清华大学出版社, 2006, 370~385.
82. Sandholm, T. and Gawor, J. (July 2003). The Globus Toolkit 3 Core – A Grid Service Container Framework [EB/OL]. http://www-unix.globus.org/toolkit/3.0/ogsa/docs/gt3_core.pdf, 2006
83. Smith RG. The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver [J]. IEEE Transactions on Computers, 1980, 29(12):1104-1113.
84. Sun Grid Engine[EB/OL].<http://www.sun.com/software/gridware/>, 2006
85. SUN Microsystem, J2EE[EB/OL].[2004].<http://java.sun.com/j2ee>.
86. Swami, B. Iyer. A polynomial time algorithm for optimizing joins queries. In Proc. IEEE Conf. on Data Engineering, Vienna, Austria, April 1993: 345-354.
87. Tomasic A, Raschid L, Valduriez P. Scaling access to distributed heterogeneous data sources with DISCO. IEEE Transactions on Knowledge and Data Engineering 10, 5 Oct. 1998:808~823
88. W Allcock, A Chervenak, I Foster et al. The data grid: Towards architecture for the distributed management and analysis of large scientific datasets. Journal of Network and Computer Applications, 2000, 23(3): 187~200.
89. Wasson, G., Beekwilder, N., Morgan, M., Humphrey, M. (2004). OGSI.NET: OGSI-Compliance on the .NET Framework. Proceedings of the 4th IEEE/ACM International Symposium on Cluster Computing and the Grid. Chicago, Illinois. CS Press, 2004.
90. Web services, Web Services Activity [EB/OL]. <http://www.w3.org/2002/ws/>, 2006
91. Welch V., Siebenlist F., Foster I., Bresnahan J., Czajkowski K., Gawor J., Kesselaman, C. Meder, S. (2003). Security for Grid Services[EB/OL].[2006].<http://www.globus.org/security/GSI3/GT3-Security-HPDC.pdf>.
92. Xiangyang Chen, Honghua Tan, Xiulian Ju, 北京: 计算机网络与通信[M].清华大学出版社, 2005.32~52
93. Yanda Li, 信息科学技术概论[M]. 清华大学出版社, 2005. 203~205, 214~219
94. Zhang, W., Zhang, J., Ma, D., Wang, B., and Ched, Y. Key Tech-nique Research on Grid Mobile Service[R]. Proceedings of the 2nd International Conference on Information Technology for Application, Harbin, China. 2004

致 谢

值此论文完成之际，首先我要把真挚的谢意献给我的导师王文生研究员。感谢王老师在我攻读硕士学位期间，在学习中给予我孜孜不倦的教诲，在工作中给予我高屋建瓴的指导，在生活中给予我无微不至的关怀。本文从开始选题，资料收集、开题、研究和撰写的每一个环节，无不得到王老师悉心指导与帮助。

感谢中国农业科学院网络中心的谢能付、杨晓蓉、赵维夷、王丹、杨勇等各位老师，在学习、工作和生活的各个方面都给予我很大的帮助和关怀，为我提供了良好的工作和学习环境，在思路的开拓上给了我很大的帮助。

感谢我的师兄师姐，项颖、李期位；

感谢我的同学，朱海鹏、郦晶、刘升平、寇远涛，带给我三年的精彩生活；

感谢我的舍友，李蕾、韩洁、李好，归属各异却同样杰出的“四个小兵”；

感谢我的朋友，杨慧、孙丽娜、刘盈斐、孙敬等，心领神会的“君子之交”；

感谢世界银行的同事，Xiaoqing、Lily、Liping、Shuo、Shiyong，感谢他们的宽容与帮助；

感谢敬爱的王馥玲主任，让我对生活和工作充满信心；

感谢可爱的凯琳和库比，是他们让我懂得快乐的真谛；

我还要向那些百忙中审阅我论文的老师表示深深的谢意。感谢参考文献中的所有作者，是您为我指点迷津，让我求知真理，热爱科研。

最后，我要感谢我的家人。正是你们默默无闻的奉献和一贯的支持和鼓励，才使我有信心去克服一切困难，有充足的时间和精力去完成学业。你们全心的付出和无言的关爱始终是我前进的巨大动力。没有你们的支持，也就没有本文的开始！

人生驿站，挥别在即，祝愿人生美好！

郭曼

2007年6月9日

作者简历

郭曼，女，1982 年 5 月出生于山东临沂。2004 年 9 月考入中国农业科学院研究生院，主要从事基于数据网格的农业数据库集成研究。攻读硕士学位期间，发表论文三篇。

- [1] Guo Man, Zhu Haipeng. Ontology-Based Agriculture scientific data integration in OGSA-DAI Environment[C]. Asia Federation of Information Technology in Agriculture 2006, 11-Nov-2006.
 - [2] 郭曼, 朱海鹏, 郦晶. 基于 OGSA 的农业科技数据库集成研究[J]. 农机化研究, 2007 (10) .
 - [3] 王文生, 郭曼等. 信息技术在农业立体污染防控的应用[J]. 农业信息网络, 2005(12).
-