

分类号 _____

密级 _____

UDC _____

编号 _____

华中师范大学 硕士学位论文

虚拟天文台数据访问服务(VO-DAS)之任务调度研究 及VO-DAS的应用

田海俊

指导教师 郑小平教授、赵永恒研究员、崔辰州副研究员

华中师范大学物理科学与技术学院

申请学位级别 硕士 学科专业名称 理论物理

论文提交日期 2007年6月 论文答辩日期 2007年6月

培养单位 物理科学与技术学院

学位授予单位 华中师范大学

答辩委员会主席 _____

Job Scheduling Research in Virtual Observatory Data Access Service (VO-DAS) and VO-DAS Application

Hai-Jun Tian

Supervisor:

Prof. Xiao-Ping Zheng & Prof. Yong-Heng Zhao & Dr. Chen-Zhou Cui

HuaZhong Normal University

May, 2007

*Submitted in total fulfilment of the requirements for the degree of Master
in Theory of Physics*



摘 要

天文观测数据资源具有时间跨度大、数据量大、存储管理分散、管理工具驳杂等特点。如何提供给天文学家一个统一访问这些分布存放的异构数据资源的方案,是虚拟天文台的一个重要研究课题。

计算机与互联网技术的飞速发展,网格技术、XML技术、语义网技术等全新IT技术的涌现,以及在此技术背景下,国际虚拟天文台联盟(IVOA)依据天文自身的特点不断提出并完善的各种规范标准,使得海量、分布式、多波段天文数据的无缝融合和处理成为可能。

对于异地异构数据的统一访问,我们基于开放网格服务架构(OGSA)提出了一种网格的解决方案:使用OGSA-DAI技术实现了对异地异构的天文星表数据、图像数据和光谱数据的统一封装(DataNode);利用ADQL语言完成对任务的统一描述;基于WSRF框架完善了对数据资源、计算资源以及存储资源的任务调度。我们设计的虚拟天文台数据访问服务(VO-DAS)实现了对数据资源、计算资源、存储资源的自动发现以及异地异构数据的联合访问并对访问结果进行数据分析的一体化工作模式,这将使天文数据源的多波段交叉认证、海量数据分析及对分析结果的可视化等成为可能。VO-DAS支持国际虚拟天文台联盟(IVOA)的各项相关标准,使得它具有良好的互操作性,它的对外接口简单实用、可以针对不同需求的天文数据用户发展出多种网格应用产品。

论文以VO-DAS的任务调度及其实现为重点,分别对VO-DAS的设计模式、Session机制、生命周期、资源销毁、异常处理等模块进行了详细的阐述,并从多个角度分别给出了系统的设计图。为了验证以OGSA-DAI为基础的天文数据访问的可行性和性能,我们采用两个科学范例对VO-DAS原型进行了实验。

论文最后以VO-DAS对China-VO Ephemeris WS 计算平台的扩展为例,介绍了VO-DAS外部接口的扩展方法,以及VO-DAS在星历计算方面的应用,并简要阐述了VO-DAS在其他方面的科学应用。

关键词: 虚拟天文台, 数据访问, 网格技术, OGSA-DAI



Abstract

The resources of observed data in astronomy have such characters as large time span, great quantity of data, break-up storage and management, heterogeneous management tool, and so on. An important research subject of the virtual observatory is about how to provide the astronomers with a programme on visiting these resources stored in different organisms.

The great development of the computer and the internet technology, the emerge of the totally up-to-date IT technology, such as grid technology, XML technology, semantic network technology, and so on, and on this technology background, according to the character of astronomy itself, the International Virtual Observatory Alliance(IVOA) continuously put forward and make them perfect various standard document, this make it possible the merge monolithic of the astronomy data which great capacity, break-up and multiple wave band.

As for consolidated visit to the resources stored in different organisms, basing on OGSA, we put forward a solution programme for the grid: making use of the OGSA-DAI technology, we can carry out the DataNode of the astronomy data about the surface of the stars, the picture data and the spectrum data in different organisms; taking advantage of ADQL language, we can go through with the description of the whole task; grouding on WSRF structure, we can make it perfect in dispatching tasks of the data resources, caculation resources and storage resources. The VO-DAS that we have designed bring into effect the close-coupled work pattern: auto-discover the data resources, caculation resources and storage resources, visite the resources stored in different organisms and data-analyse the result , this will make it possible the multiple wave band cross-authentication, the data-analysis and the visualisation of the analysis result. VO-DAS support all the related standards of the IVOA, this makes it have excellent operability, its external interface is simple and practical, it can develop various grid application products aimed at different demands of different astronomy data clients.

Focusing on diapatching VO-DAS tasks and bringing it into effect, we elaborate on the modules of VO-DAS, such as its design mode, Session organisms, life cycle, resources destroying, exception handling, and so on, and put forward the blue prints of the system indifferent angles. To verify the feasibility and performance of the visit



to the astronomy data basing on OGSA-DAI, we adopt two scientific model to test the prototype of VO-DAS.

Take for example the expansion of the VO-DAS to the caculation platform of the China-VO Ephemeris WS, we introduce the expansion method of VO-DAS' exterior interface, and VO-DAS' application in the caculation of the star calendar, and we also elaborate on VO-DAS' application in the scientific applications in the other fields.

Keywords: Virtual Observatory, Data Access Service, Grid, OGSA-DAI

目 录

目录	iv
第一章 引言	1
1.1 虚拟天文台研究背景 [1]	1
1.2 国内外研究现状 [3]	3
1.3 VO-DAS的研究意义	4
1.4 论文组织结构	4
第二章 关键技术和概念综述	6
2.1 网格技术	6
2.1.1 开放式网格体系结构(OGSA) [8]	6
2.1.2 OGSA的基础访问结构	6
2.1.3 开放式网格服务基础结构(OGSI) [13]	8
2.1.4 Web服务资源框架(WSRF) [14]	9
2.2 网格技术在国内外现状	11
2.3 网格相关的中间件	12
2.3.1 Globus Toolkit [15]	12
2.3.2 OGSA-DAI [16]	13
2.4 国际虚拟天文台联盟(IVOA)及其相关规范	17
2.4.1 IVOA天文数据查询语言(ADQL) [17]	17
2.4.2 简单图像访问规范(SIA) [18]、简单光谱访问规范(SSA) [19]	19
2.4.3 FITS、VOTable等天文格式规范	20
第三章 基于WSRF的数据访问服务(VO-DAS)	22
3.1 VO-DAS的需求分析	22
3.2 VO-DAS的框架 [24]	23
3.2.1 VO-DAS五种访问接口	23
3.2.2 VO-DAS两大核心模块	23



3.2.3	天文数据查询语言 (ADQL)	25
3.2.4	数据查询	26
3.2.5	数据分析	27
3.2.6	数据分发	27
3.2.7	服务的信息注册与信息发现	27
3.2.8	数据格式	27
3.2.9	VO-DAS的运行环境	27
3.2.10	VO-DAS的性能	27
第四章	VO-DAS任务调度研究与实现	29
4.1	多任务调度的几种可能方案	29
4.2	VO-DAS任务调度方案的设计	29
4.2.1	VO-DAS的Session机制	29
4.2.2	VO-DAS的生命周期控制	30
4.2.3	VO-DAS的资源销毁	31
4.2.4	数据查询设计	32
4.2.5	数据的分析	35
4.2.6	系统的状态监控	36
4.2.7	VO-DAS异常处理	37
4.2.8	VO-DAS安全机制	37
4.3	VO-DAS任务调度的实现和测试	38
4.3.1	结构图 (以三个DataNode为例)	38
4.3.2	类图	38
4.3.3	序列图	38
4.3.4	测试方案和测试结果	39
4.4	和国外同类虚拟天文台系统的比较	40
第五章	VO-DAS的功能扩展及其应用	48
5.1	基于VO-DAS平台可视化日食带的中心食带	48
5.1.1	China-VO Ephemeris WS 计算平台 [31]	48
5.1.2	VO-DAS的功能扩展	50
5.1.3	通过VO-DAS可视化中心食带	52
5.2	VO-DAS其他应用	53



第六章 总结与展望	54
附录 A JDL的xml表述形式	55
参考文献	57
发表文章目录	60
致谢	61

表 格

1.1 国际各国虚拟天文台项目	3
4.1 VO-DAS的部分异常列表	38

插 图

2.1	Web Services 结构图	7
2.2	一个典型的Web Services 调用过程	8
2.3	从WSDL中生成stubs文件	9
2.4	GT4 结构图	13
2.5	数据服务资源DSR、数据资源访问门DRA以及数据资源DR（以关系型数据资源为例）三者之间的关系	15
2.6	一次SQL查询，OGSA-DAI各模块间的交互	15
2.7	OGSA-DAI 体系结构图	16
3.1	VO-DAS设计图	24
4.1	VO-DAS Session机制	30
4.2	同步查询工作流	33
4.3	异步查询工作流	33
4.4	联合查询工作流	34
4.5	VO-DAS 结构图（以三个DataNode为例）	41
4.6	VO-DAS 类图	42
4.7	VO-DAS的启动	43
4.8	VO-DAS的关闭	43
4.9	VO-DAS的元数据查询	44
4.10	VO-DAS的同步查询	44
4.11	VO-DAS的异步查询	45
4.12	VO-DAS上传数据	46
4.13	VO-DAS的数据分析	47
5.1	China-VO Ephemeris WS功能结构图	50
5.2	Activity生命周期的序列图	51
5.3	2008年03月12日之后，第一次中心食食带	52
5.4	2006年08月12日之前，最后一次中心食食带	53

第一章 引言

1.1 虚拟天文台研究背景 [1]

四百年前伽利略首次把望远镜指向太空，使人类摆脱了仅能用肉眼直接观测太空的历史，为从哥白尼开始的天文学革命提供了大量的科学证据。历史悠久的天文学经过哥白尼、伽利略、开普勒和牛顿等人的发展，演变成了一门崭新的科学，同时也催生了现代科学技术。

到一百五十年前，由于照相技术和光谱技术在天文观测中的应用，用人眼作为唯一的天文探测器的时代结束，诞生了天文学的新分支—天体物理学，并发展成为现代天文学的主流。

五十多年前，在第二次世界大战中得到蓬勃发展的无线电技术使得人类的视野跃出了可见光的波段，发展成为射电天文学。之后不久随着宇航时代的到来，空间天文学诞生，天文观测不再局限于地面。人类对宇宙的观测范围扩展到了 γ 射线、X射线、紫外和红外波段。天文学开始进入全波段天文学时代。

从20世纪90年代开始，天文学又经历着革命性的变化。这一变化是由前所未有的技术进步所推动的，即望远镜的设计和制造技术、大尺寸探测器阵列的设计和制造技术、高性能计算技术和互联网技术。

随着众多先进的地面与空间天文设备的投入使用，特别是大规模CCD探测器的使用，使得观测数据量急速增长。例如目前哈勃空间望远镜（HST）每天大约产生5GB的数据；我国正在建造的大天区面积多目标光纤光谱望远镜（LAMOST [2]）也将每天至少产生3GB到5GB的数据；而美国计划建造的“大口径综合巡天望远镜（LSST）”，又称为“暗物质望远镜”，每天的观测数据将达到18TB的量级！

如此巨大的数据产出，在天文学历史上第一次使天文学家得到的数据多得用不了，使天文学进入一个数据富庶时代。以往那个辛辛苦苦观测许久但数据还是不够用的年代一去不复返。

随着天文望远镜及终端设备的设计与制造技术的不断提高，天文观测能力大大增强，天文学已从古老的光学观测变为全波段的天文学，并正在进入一个“数据雪崩”时代。

天文信息爆炸式增长不仅给天文学家带来了天文发现的巨大机遇，也同时带来了数据存储、访问和处理等方面的巨大挑战。传统的借助人工来对数据进行分析处理的方式已经不能满足海量的天文数据了。上百年的天体观测资料积累造成了这样一种情况：世界各国的天文资料保存和管理方式存在巨大差异，不同历史时期数据



保存和管理方式也存在巨大差异。虽然世界各地的天文数据是自由共享的,但是数据结构的混乱和数据存储的多样性就像语言不通一样,严重影响着天文学家对这些数据资料的使用效率。

幸运的是,计算机与互联网技术的飞速发展,网格技术、XML技术、语义网技术等全新IT技术的涌现,使得海量、分布式、多波段天文数据的无缝融合和处理成为可能。

在这样的背景下,旨在将世界范围内主要天文研究资源无缝透明地整合在一起的虚拟天文台(即virtual observatory,简称VO)设想应运而生。

如果说利用 γ 射线巡天、X射线巡天、紫外巡天、光学巡天、红外巡天和射电巡天所得到的观测数据,用适合的方法对数据进行统一规范的整理、归档,便可以构成一个全波段的数字虚拟天空;而根据用户要求获得某个天区的各类数据,就仿佛是在使用一架虚拟的天文望远镜;如果再根据科学研究的要求开发出功能强大的计算工具、统计分析工具和数据挖掘工具,这就相当于拥有了虚拟的各种探测设备。这样,由虚拟的数字天空、虚拟的望远镜和虚拟的探测设备所组成的机构便是一个独一无二的虚拟天文台。由此可见,VO是在互联网时代里天文学发展的必然产物。

VO将使天文学取得前所未有的进展,它将成为开创“天文学发现新时代”的关键性因素。虚拟天文台将是独一无二的,它将TB甚至PB量级的数据库、波长遍及从 γ 射线到射电波段的数十亿个天体的图像库、高度复杂的数据挖掘和分析工具、具有数千PB量级容量的存储设备和每秒运算次数达到万亿次的超级计算设备、以及各主要天文数据中心之间的高速网络连成一体;它使世界各地的天文学家可以快速查询各个PB量级大小的数据库;使埋藏在庞大星表和图像数据库中的多变量模式可视化;增加发现复杂规律和稀有天体的机会;鼓励研究团体间的实时合作;允许进行大规模的统计研究,首次使数据库的内容可以和复杂精密的数值模拟结果进行对比。虚拟天文台将促进我们对许多决定宇宙演化的天体物理过程的理解。它会更经济的投资产生新的和更好的科学。虚拟天文台将作为一个协调性的和操作性的机构来促进新型的工具、协议和合作方面的发展,以充分实现现代天文数据库的科学潜能,从而将成为“天文学发现”的推进器。

VO的好处是显而易见的,主要表现在如下几个方面:

1. 天文望远镜等观测设备得到的数据能被不同的用户出于不同的目的重复使用,提高了昂贵的观测设备的科学效益;
2. 观测数据按照统一的方式管理起来有利于长期的保存和利用,使科学数据的价值最大化;
3. VO可被全球各种各样的群体访问,包括那些没有经济能力建造和运行大型观测设备的群体,能大大促进发展中国家和不发达国家的天文研究;



表 1.1: 国际各国虚拟天文台项目

项目	地区	网址
中国虚拟天文台(China-VO)	中国	http://www.china-vo.org
国家虚拟天文台(NVO)	美国	http://www.us-vo.org
英国虚拟天文台(AstroGrid)	英国	http://www.astrogrid.org
欧洲虚拟天文台(Euro-VO)	欧洲	http://www.euro-vo.org
法国虚拟天文台(VO-France)	法国	http://www.france-vo.org
加拿大虚拟天文台(CVO)	加拿大	http://services.cadc-ccda.hia-ihp.nrc-cnrc.gc.ca/cvo
印度虚拟天文台(VO-India)	印度	http://vo.iucaa.ernet.in
德国虚拟天文台(GAVO)	德国	http://www.g-vo.org/portal
澳大利亚虚拟天文台(Aus-VO)	澳大利亚	http://www.aus-vo.org
日本虚拟天文台(JVO)	日本	http://jvo.nao.ac.jp
俄罗斯虚拟天文台(RVO)	俄罗斯	http://www.inasan.rssi.ru/eng/rvo
意大利虚拟天文台(DRACO)	意大利	http://www.as.oat.ts.astro.it/twiki/bin/view/Draco/WebHome
韩国虚拟天文台(KVO)	韩国	http://hvo.elte.hu/en
匈牙利虚拟天文台(HVO)	匈牙利	http://www.france-vo.org
西班牙虚拟天文台(SVO)	西班牙	http://laeff.esa.es/svo
亚美尼亚虚拟天文台(ArVO)	亚美尼亚	http://www.aras.am/arvo.htm

4. VO是非常好的公众教育设施,VO是面向全球、全社会的。它让普通公众和天文学家一样都能接触到真实的天文资源和服务。这会激发大众对天文学的兴趣,提高大众的天文水平,促进天文学的普及。

1.2 国内外研究现状 [3]

虚拟天文台的概念提出后各国也相继出台了各自的虚拟天文台计划。为了相互借鉴,相互促进以及国际上各项目之间的协调与合作,2002年成立了国际虚拟天文台联盟(International Virtual Observation Alliance, IVOA [4]),目前国际虚拟天文台联盟中成员已经增加到16个国家。如表1.1所示:

近年来,各国虚拟天文台项目分别提出了自己的数据统一访问方案。比如,英国实现了世界上第一个结构比较完整的虚拟天文台Astrogrid [5];美国国家虚拟天文台(NVO)先后提出了SkyQuery [6]原型和OpenSkyQuery [7]原型;日本虚拟天文台(JVO)也提出了自己的数据访问方案。在这些研究的基础上,IVOA正在制



订SkyNode数据访问模型。SkyNode是基于Web Service设计的，支持异地异构的多波段天文数据的交叉证认。但是由于Web Service的限制，SkyNode仅仅能够用来对小数据集进行操作，无法完成大数据集的访问和交叉证认。

为了紧跟时代的发展，缩小与发达国家的信息天文学方向的差距，2002年以中国科学院国家天文台为首在国内天文界首先提出建设“中国虚拟天文台”项目。同年十月，中国虚拟天文台加入国际虚拟天文台联盟。经过近几年的研究探索，中国虚拟天文台依据国际虚拟天文台联盟（IVOA）的各项相关标准，使用OGSA-DAI技术实现了对天文星表数据、图像数据和光谱数据的封装（DataNode），利用Globus Toolkits 4（GT4）等网格技术，提出并初步实现了自己的数据访问平台（VO-DAS）。

1.3 VO-DAS的研究意义

中国虚拟天文台提出的数据访问平台VO-DAS将具有以下功能特点：

- 能够访问分布异地、异构天文数据库；（初步实现）
- 支持自动发现数据资源；（初步实现）
- 支持特定空间区域的数据访问；（初步实现）
- 支持海量数据访问；（初步实现）
- 支持天文数据的交叉证认；（初步实现）
- 支持星表数据，图像数据和光谱数据的访问；（仅实现了星表数据的访问）
- 集成数据挖掘平台；（尚未实现）
- 数据可视化研究；（尚未实现）
- 天文科普教育；
- 具有标准的数据访问接口；
- 有良好的互操作性，可以和任何符合IVOA规范的VO应用程序互联；

-具有良好扩展性，VO-DAS是基于OGSA-DAI的，OGSA-DAI这个灵活框架，决定了VO-DAS的灵活扩展性，比如可以扩展Swiss Ephemeris软件包的星历计算能力。

鉴于VO-DAS上述的功能特点，那么如果能够实现VO-DAS，将很好地解决虚拟天文台的关键问题，它也将成为中国虚拟天文台的框架和中国天文数据对外的一个窗口，尤其是我国即将实现的大天区面积多目标光纤光谱望远镜（LAMOST）产生的海量数据的对外窗口（VO-enabled LAMOST）。

1.4 论文组织结构

本文阐述了中国虚拟天文台数据访问系统（VO-DAS），着重阐述基于WSRF的VO-



DAS的任务调度，具体内容按下面顺序依次叙述：第一章为引言部分，着重介绍了VO的研究背景、目前国内外的研究现状；第二章综述了有关网格的一些关键技术和概念，以及国际虚拟天文台联盟的一些相关规范；第三章介绍了基于WSRF的数据访问服务系统的框架；第四章是本论文的重点所在，将详细介绍VO-DAS的任务调度方案。第五章主要从VO-DAS平台可视化日食带的中心食带的案例来介绍VO-DAS的应用，同时简单介绍一下VO-DAS在数据查询以及数据分析的更广泛的应用。第六章，对全文进行总结，指出本文研究工作的主要成果和今后工作的展望。

第二章 关键技术和概念综述

2.1 网格技术

网格是伴随着互联网技术而迅速发展起来的,最初是专门针对复杂科学计算应用的一种新型计算模式。随着网格技术的发展和应用的扩展,网格逐步进入到商业使用中。人们预言,网格技术和应用将成为具有高性能处理、海量数据存储和大量仪器设备等特征的21世纪人类社会的信息处理基础设施。通过它可以汇聚Internet中分散异构、动态变化的计算和信息资源,将其中不同组织和机构的资源虚拟化,使人们能像使用一台计算机那样使用Internet中的各种资源,把Internet从通讯和信息交互平台提升到资源共享和协同工作的平台。

2.1.1 开放式网格体系结构(OGSA) [8]

2000年底,美国网格论坛、欧洲网格论坛和亚洲网格论坛合并组成了全球网格论坛(Global Grid Forum, GGF)。GGF是由用户、研发人员和产品生产厂家组成的致力于网格技术标准化的社团。全球网格论坛(GGF) [9]是进行有关网格技术的标准化工作主要机构。GGF的主要工作是制订与开放式网格服务体系结构(Open Grid Service Architecture, OGSA)相关的标准。

2002年6月, Globus联盟首次在全球网格论坛(GGF)上提出了开放式网格服务体系结构(OGSA)的理念,服务网格的研究开始走上了正确的轨道。

所谓网格体系结构或网格服务体系结构是确定网格服务系统中的基本系统组件,界定和描述设置系统基本组件的目的和组件的功能,并说明组件之间如何进行交互式的操作的概念模型。与其他开放系统类似, OGSA通过定义标准的服务和接口实现虚拟组织之间相互开放。OGSA是面向服务的体系结构, 所谓“服务”在OGSA中被定义为:“在网络支持下通过信息交换能够给客户提供某种能力的实体”。在分布式环境中,从信息交换的角度来看,服务可以看作是导致服务实体执行某些操作的特定信息交换序列,因此,在OGSA中可见的操作都是信息交换的结果。OGSA将服务操作封装在一个面向消息的公共接口之中,从而实现服务的虚拟化,即将服务的定位与服务实现细节相分离。在OGSA中,计算资源、存储资源、网络、程序、数据库等都被表示为服务。

2.1.2 OGSA的基础访问结构

在网格应用的资源分布的环境中,要实现分布式网络计算必须解决好如何



在OGSA服务间实现交互的问题。为此，必须具备用机器可理解的方式描述服务需求和接口的能力，动态地发现、定位和组合网格服务的能力。

GGF选择了W3C制订的Web服务相关标准作为实现其目标的基础，并根据OGSA的需要对Web服务进行扩充与增强，构建其基础服务结构。

2.1.2.1 Web服务 [10]

Web服务与其支撑平台一道在以计算机网络为支撑的分布、异构计算机环境中提供一种通用的互操作手段，它具有动态资源发现和组合能力。由于在Web服务之上构建应用系统十分方便，它已经成为构建互联网分布式应用系统（E-Science，E-Business和E-Governance等）的基本手段。Web服务涉及四个要素：Web服务说明（WSD）、用于封装WSD的消息封（envelope）、对下层传输协议的绑定、运载消息的传输协议以及Web服务的发布机制。

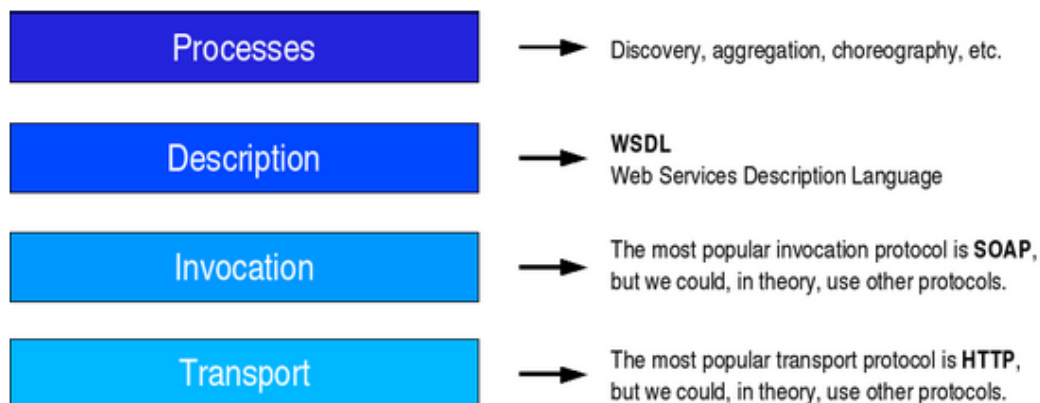


图 2.1: Web Services 结构图

1) Web服务描述语言（WSDL） [11]

WSDL（Web Service Description Language）是Web咨询机构W3C所定义的Web服务描述语言。它为描述Web服务提供模型和XML格式。WSDL对Web服务的描述分为抽象描述和具体描述，描述采用结构化方式进行，以提高服务描述的可重用性。

在抽象描述阶段，Web服务用收发的消息来描述，采用XML Schema格式。“操作”是对服务中能够收发的消息种类和消息交换格局的描述，一个操作可与一个或多个消息交换规则（消息收发序列和顺序及逻辑收发方）相联系。最后在抽象描述中，Web服务接口是成组的操作，但未与传送或具体格式挂钩。在具体描述阶段，“绑定”关系的描述将定义抽象的服务接口所使用的传送协议和具体格式；通过“可用服务端点”（Endpoint）将绑定部件与网络地址联系起来；最后将实现共同服务接口的可用服务端点合并定义为Web服务。

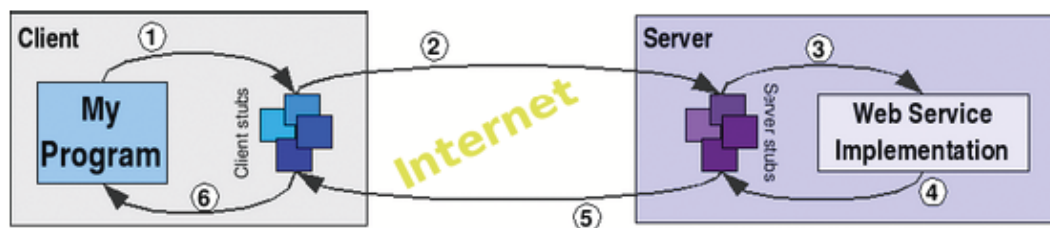


图 2.2: 一个典型的Web Services 调用过程

2) 简单对象访问协议 (SOAP) [12]

Web服务可用于在互联网上发布可用的应用程序模块（服务），完成请求服务以及WSDL和语义的协商。WSDL的描述采用符合扩展标记语言结构与约束（XML Schema）规定的文本格式，不适合直接作为应用消息传输，需要封装在固定格式的消息内再进行传输。W3C的简单对象访问协议定义一种封装格式和结构，定义了用于识别（强制性或选择性）处理部分或全部消息的对象的机制，以及如何将SOAP消息与下层传输协议绑定的方法。例如，当下层采用超文本传输协议（HTTP）时，SOAP消息可能作为HTTP POST请求与应答中的净荷（Payload）部分传输，或作为SOAP消息对HTTP GET的应答，从而实现与HTTP的绑定。为了适应远程程序调用的需要，SOAP还定义了SOAP数据模型，即用于运载RPC的数据编码结构。

3) SOAP消息的传输协议

SOAP定义了可用于装载WSDL描述的消息格式，它需要与端系统之间运载SOAP消息的传输协议相结合才能完成Web服务既定的功能。前面已经提到，应用层的超文本传输协议（HTTP）是一种可能的选择，但SOAP并不排除使用其他应用层协议，如简单邮件传输协议SMTP、微软公司的消息队列协议(Microsoft Message Queue, MSMQ)或者直接利用传送层协议（TCP）。

4) 注册或服务发现

为了便于用户查询和使用已有的服务，需要将Web Service在管理服务器上注册。相关技术和标准由“促进结构化信息标准机构”（OASIS）的通用描述、发现与集成（Universal Description Discovery and Integration, UDDI）成员组制订。Web服务的使用过程可以用图2.2来简单描述，即服务请求与提供服务双方通过“代理”（Stubs文件）交换Web服务描述（WSDL）和语义达到服务协商一致来完成的。

客户端和服务器的Stubs文件都是通过WSDL文件产生的，如图2.3

2.1.3 开放式网格服务基础结构（OGSI） [13]

在GGF进行OGSA相关文本的制订过程中，一方面认定Web服务时提供网

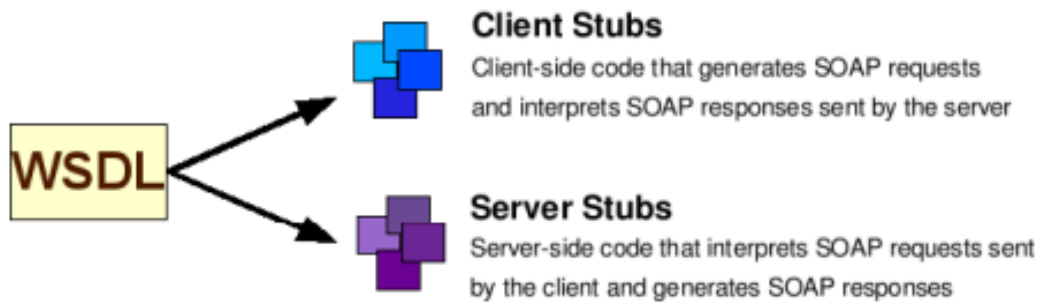


图 2.3: 从WSDL中生成stubs文件

格服务的最佳选择，另一方面又感到1.0版的Web服务还存在一些问题，不能满足OGSA的需要，需要对之进行扩展。

因此，在OGSA刚提出不久，GGF及时推出了开放网格服务基础架构（Open Grid Service Infrastructure, OGSI）来对Web服务进行扩展。

OGSI是作为OGSA核心规范提出的，其1.0版于2003年7月正式发布。OGSI规范通过扩展Web服务定义语言WSDL和XML Schema的使用，来解决具有状态属性的Web服务的问题。它提出了网格服务的概念，并针对网格服务定义了一套标准化的接口，主要包括：服务实例的创建、命名和生命期管理，服务状态数据的声明和查看，服务数据的异步通知，服务实例集合的表达和管理，以及一般的服务调用错误的处理等。

但近一两年的实践证明，OGSI存在明显的不足。其过分强调网格服务和Web服务的差别，导致了两者之间不能更好地融合在一起。由于OGSI单个规范中的内容太多，所有接口和操作都与服务数据有关，缺乏通用性，而且OGSI规范没有对资源和服务进行区分。OGSI使用目前的Web服务和XML工具不能良好工作，其过于采用XML模式，比如xsd:any基本用法、属性等，这种性质可能造成移植性差的问题。OGSI通过封装资源的状态，将具有状态的资源建模为Web服务，这种做法引起了“Web服务没有状态和实例”的争议，同时某些Web服务的实现不能适应网格服务的动态创建和销毁。另外，网格服务的定义语言GWSDL不能作为可支持Web服务描述语言WSDL 1.1的功能扩展，由于WSDL 2.0发布的延迟使之很难支持OGSI定义。

2.1.4 Web服务资源框架（WSRF） [14]

对应开放网格服务基础架构OGSI 1.0版的推出，并试图解决OGSI和Web服务之间存在的矛盾，Web服务资源框架WSRF被提了出来。2004年3月，IBM、BEA与微软联合发布了WS-Addressing协议。基于该协议规范，Globus联盟和IBM迅速推出了Web服务资源框架WSRF。结构信息标准化促进组织（OASIS）随即成立了两个技



术委员会，分别是网络服务资源框架技术委员会（WSRF TC）和网络服务通告技术委员会（WSN TC）。

WSRF采用了与网格服务完全不同的定义：资源是有状态的，服务是无状态的。为了充分兼容现有的Web服务，WSRF使用WSDL 1.1定义OGSI中的各项能力，避免对扩展工具的要求，原有的网格服务已经演变成了Web服务和资源文档两部分。WSRF推出的目的在于，定义出一个通用且开放的架构，利用Web服务对具有状态属性的资源进行存取，并包含描述状态属性的机制，另外也包含如何将机制延伸至Web服务中的方式。

WSRF的规范是针对OGSI规范的主要接口和操作而定义的，它保留了OGSI中规定的所有基本功能，只是改变了某些语法，并且使用了不同的术语进行表达。成立网络服务通告技术委员会的目的在于定义多项规格，并进行以通知模式作为网络服务互通方式的标准化工作。利用通知模式，网络服务在传播信息给其它网络服务时，不必预先知道这些网络服务。该委员会将修订Wcation Framework的相关规格，并发布WS-NotificationPolicy规格，作为详细描述通告的相关政策语言。WSRF使Web服务体系结构发生了以下2点演变：

- ◆提供了传输中立机制来定位Web服务；
- ◆提供获取已发布服务的信息机制集，具体的信息包括WSDL描述、XML模式定义和使用这项服务的必要信息。

和OGSA的最初核心规范OGSI相比，WSRF的优势表现为如下5点：

- ◆融入Web服务标准，同时更全面地扩展了现有的XML标准，在目前的开发环境下，使其实现简单化。

◆OGSI中的术语和结构让WS组织感到困惑，因为OGSI错误地认为Web服务一定需要很多支撑的构建。WS-Resource Framework通过对消息处理器和状态资源进行分离来消除上述隐患，明确了其目标是允许Web服务操作对状态资源进行管理和操纵。

- ◆OGSI中的Factory接口提供了较少的可用功能，在WS-Resource Framework中定义了更加通用的WS-Resource Factory模式。

◆OGSI中的通知接口不支持通常事件系统中要求的和现存的面向消息的中间件所支持的各种功能，WS-Resource Framework中规范弥补了上述的不足，从广义角度来理解通知机制，状态改变通知机制正是建立在常规的Web服务的需求之上。

◆OGSI规范的规模如此庞大，使读者不能充分理解其内容，以及明确具体任务中所需的组件。在WS-Resource Framework中通过将功能进行分离，使之简化并拓展了组合的伸缩性。



作为OGSA最新核心规范的Web服务资源框架WSRF,它的提出加速了网格和Web服务的融合,以及科研界和工业界的接轨。

OGSA和WSRF目前都处于不断的发展变化之中。2004年6月,OGSA 1.0版本发布,阐述了OGSA与Web服务标准的关系,同时给出了不同的OGSA应用实例。OGSA 2.0版本于2005年6月发布。作为WSRF的基础支撑协议WS-Addressing,也得到了万维网联盟(W3C)的承认,并成立了W3C WS-Addressing工作组。

对于WSRF本身而言,由于其提出不久,其规范还有待在实践中得到进一步应用证明,并逐步得到完善。基于OGSA和WSRF的服务网格平台和规范协议,将最终成为下一代互联网的基础设施,所有的应用都将在网格的基础平台上得以实施。

2.2 网格技术在国内现状

美国政府自20世纪90年代就开始投资于网格技术研究,累计用于网格技术基础研究方面的经费已近5亿美元,相继提出了各种网格理论,并建立以Globus、Globe、NetSolve、Javalin为代表的实验模型和项目。作为网格计算最杰出的代表,由美国Argonne国家实验室实施研发的Globus,被认为是网格技术的典型代表。Globus通过对网格技术的研究相应软件的开发及相关标准的制定,力图构造一个规范的网格计算环境,实现对高性能计算机远程计算资源普遍、可靠、一致性的访问。在对资源管理、安全、信息服务及数据管理等网格计算的关键理论进行研究的同时,Globus于1999年开发出的能运行于各类平台的网格计算工具软件GlobusToolkit是该系统最重要的成果,其源代码完全开放,可以帮助规划和组建大型网格实验平台,开发适合大型网格系统运行的大型应用程序。目前,Globus的技术已在NASA网格、欧洲数据中心、美国国家技术网格等8个项目中得到应用。美国国防部规划实施的“全球信息网格”计划,将在2020年完成。作为该计划的一部分,美国海军和海军陆战队已先期启动了一个耗资160亿美元、历时8年的项目,进行系统的研制、建设、维护和升级。

英国政府已决定投入1亿英镑用于网格研究,建立英国的国家网格项目,形成全国范围内互联互通的网络系统。欧盟已于2001年1月起,正式实施由欧洲粒子实验室等6个研究机构参与的数据网格项目,计划在3年内投资980万欧元,采用Globus网格技术,建立一个在欧洲范围内不同数据源之间实现海量数据计算与数据交换的大规模分布计算环境和共享平台,为科学研究提供强大保障。法国启动的国家网格计划,前3年计划的经费投入是1000万美元。荷兰国家网格计算计划将使5个大学的研究人员能够更有效地在生物信息到粒子物理等科研项目方面进行合作。这一网格计算计划包含5台Linux群集系统(每个大学各一台),通过荷兰大学的高性能网络-SUPERNet连接在一起。



我国已经完成的网格研究项目主要有清华大学的先进计算基础设施ACI和以中科院计算所为主的国家高性能计算环境NHPCE。目前正在进行的网格研究项目主要有：“863”专项计划支持的“中国国家网格（CNGrid）”项目；教育部“十五”211工程公共服务体系建设的“中国教育科研网格（ChinaGrid）”；由中国科学院计算所领衔开发的“织女星网格”项目；由上海多所大学参加的“上海教育科研网格”项目；由航天总公司第二研究院和清华大学共同开展的“仿真网格”项目。其中，由中国科学院计算所领衔攻关的“织女星网格”项目，涉及到网格超级服务器、网格操作系统、信息网格和知识网格的架构、性能、应用的各个层面，其未来的目标是具备大规模的数据处理能力、高性能计算能力，以及资源共享和提高资源利用率方面的能力。

2.3 网格相关的中间件

2.3.1 Globus Toolkit [15]

怎样将分布在不同地方的计算机组合成一个网格呢？以前大家用的最多的工具就是Globus Toolkit工具包。由全球网格论坛（GGF）下属Globus项目组成员联合开发的Globus Toolkit标准工具包，已被公认为当前建立网格系统和开发网格软件事实上的参考标准。Globus项目是国际上与网格计算相关的最有影响力的项目。

Globus Toolkit之所以能取得成功的关键是它选择了开放系统源码的道路，因而得到了众多公司的支持。它基于开放结构、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随着体系结构的变化也经历了几次飞跃，变得越来越完善。自1997年起，Globus Toolkit工具包的第二版（GT2）成为了网格计算的事实标准。它着重于可用性和互操作性能力，定义和实现了一些协议、API和服务。

2002年，Globus项目组推出了一个全新的网格标准OGSA—开放网格服务体系，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。而符合OGSA规范的Globus Toolkit 3.0（GT3）随后推出，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。

2004年，公布了统一网格计算和Web服务的新标准“WS-Notification”和“WS-Resource Framework”（Web服务资源框架）。WSRF是OGSI的重构和发展，利用了新的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的能力。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年年初，Globus Toolkit 4（GT4）发布，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（如图2.4）。从中间件的角度看，WSRF提供了应用间的简单的互连互通。自此，WSRF成为网格中间件事实上的“国际标准”。

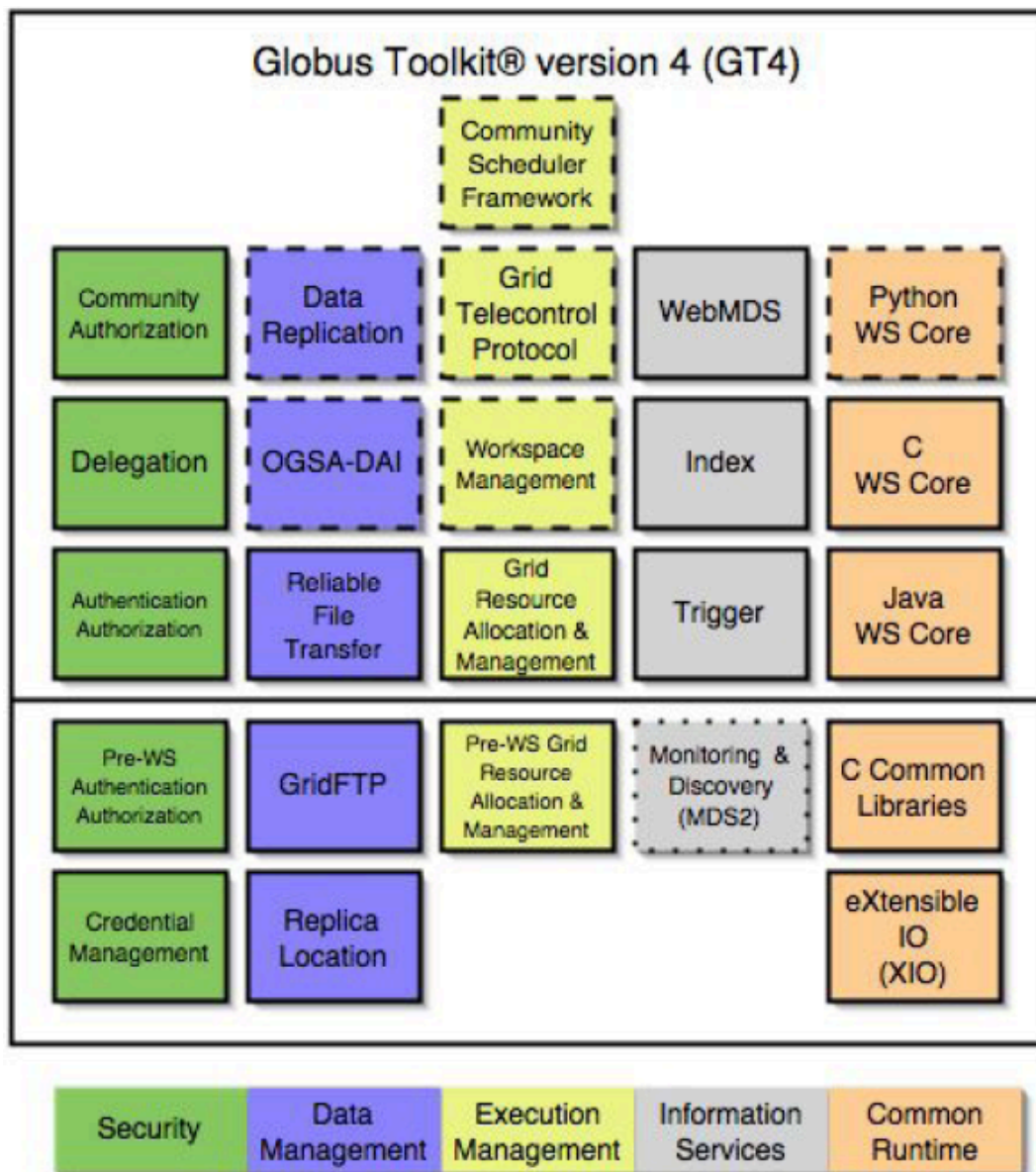


图 2.4: GT4 结构图

2.3.2 OGSA-DAI [16]

OGSA-DAI 是一种中间件，其设计目标是提供一种简便的方法，在网格环境中实现数据的访问和集成。在网格环境下OGSA-DAI 主要有以下功能特点：



- 1.支持各种数据资源以网格的方式发布,主要包括:关系型数据库、XML数据库以及通过Web Services发布的各种文档数据等常用的数据资源。
- 2.支持通过Web服务来对各种数据资源进行查询、更新、转换以及传输等操作。
3. 支持数据的联合查询以及分布式查询。
- 4.支持数据转换(比如通过XSLT),数据的压缩、解压(比如通过ZIP或者GZIP)。
- 5.数据可以很方便地传输到客户端、其他的OGSA-DAI服务上、一些指定的URLs、FTP服务器、GridFTP服务器上、或者写入指定的文档中。
- 6.对OGSA-DAI的Web服务请求都有一个统一的格式,无论数据资源发布在哪个服务器上。
- 7.客户端可以获取OGSA-DAI的Web服务上的数据资源信息以及数据资源支持的功能特性。
- 8.OGSA-DAI的使用者可以通过扩展OGSA-DAI的Web服务来发布自己的数据资源,除了OGSA-DAI提供的功能外,同时支持该数据资源本身的功能特性。

2.3.2.1 OGSA-DAI的基本概念

(1) 数据服务(Data Services, DS)

OGSA-DAI 提供两种类型的数据服务接口: WS-RF数据服务和WS-I数据服务。一个数据服务用来配置各种数据服务资源(Data Service Resources)。

(2) 数据服务资源(Data Service Resources, DSR)

数据服务资源承担着OGSA-DAI的核心功能。它从数据服务(DS)上接受执行文档(Perform Documents),然后解析并使执行文档生效,继而顺序执行执行文档中“承载”的每一个行为指令(Activity),最后生成相应文档(Response Document)。

(3) 数据资源(Data Resource, DR)

OGSA-DAI的数据资源(DR)主要包括三种类型:关系数据库(如:MySQL, SQLServer, Oracle等)、XML 数据库(如: eXist, Xindice)、文件(如:OMIM, SWISS-PROT, EMBL等类型)。

(4) 数据资源访问“门”(Data Resource Accessor, DRA)

数据资源访问“门”是OGSA-DAI的一个扩展点,你可以针对不同的数据资源(DR)书写与该资源类型对应访问“门”。这也是OGSA-DAI封装异构数据的关键所在。DR、DRA以及DSR三者之间的关系(如图2.5所示)。

(5) 执行文档(Perform Document, PD)

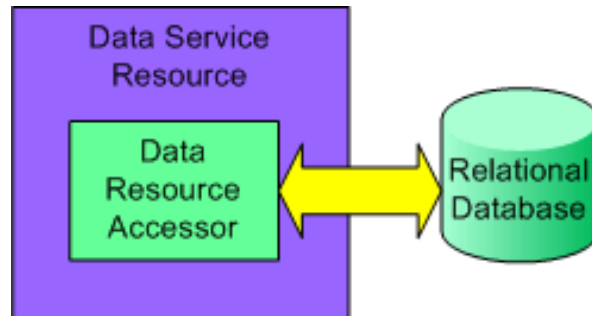


图 2.5: 数据服务资源DSR、数据资源访问门DRA以及数据资源DR（以关系型数据资源为例）三者之间的关系

一种XML格式的文档，用于定义要在DS上执行的活动，如一条SQL查询，然后再定义如何将查询的结果传送给第三方。

(6) 响应文档(Response Document, RD)

一种XML格式的文档，是DS处理执行文档后返回的结果。

(7)行为指令(Activity或Activities)

它代表客户端向OGSA-DAI下达一个或多个指令操作，比如查询或更新某数据库、将查询结果文件压缩、存入某ftp等等单元操作。它实现程序复杂功能的单元模块。

以客户端下达一次SQL查询为例，来分析各基本概念之间的调度关系，如图2.6所示：

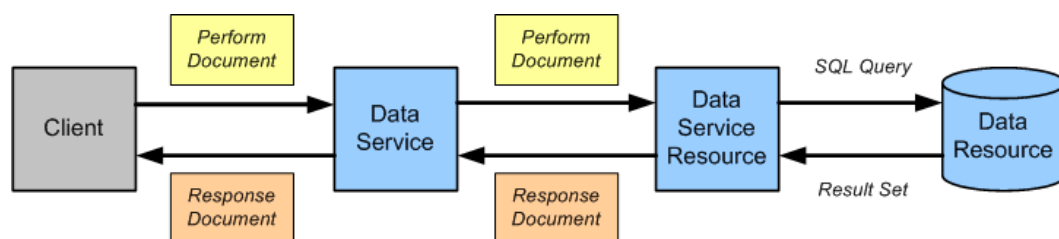


图 2.6: 一次SQL查询，OGSA-DAI各模块间的交互

2.3.2.2 OGSA-DAI体系结构

OGSA-DAI提供了将现有数据资源,如关系数据库和XML (eXtensible Markup Language)数据库以及文件目录集成到网格环境中的基本架构.它主要包括四个层次(如图2.6):数据层、业务逻辑层、表示层和客户层。

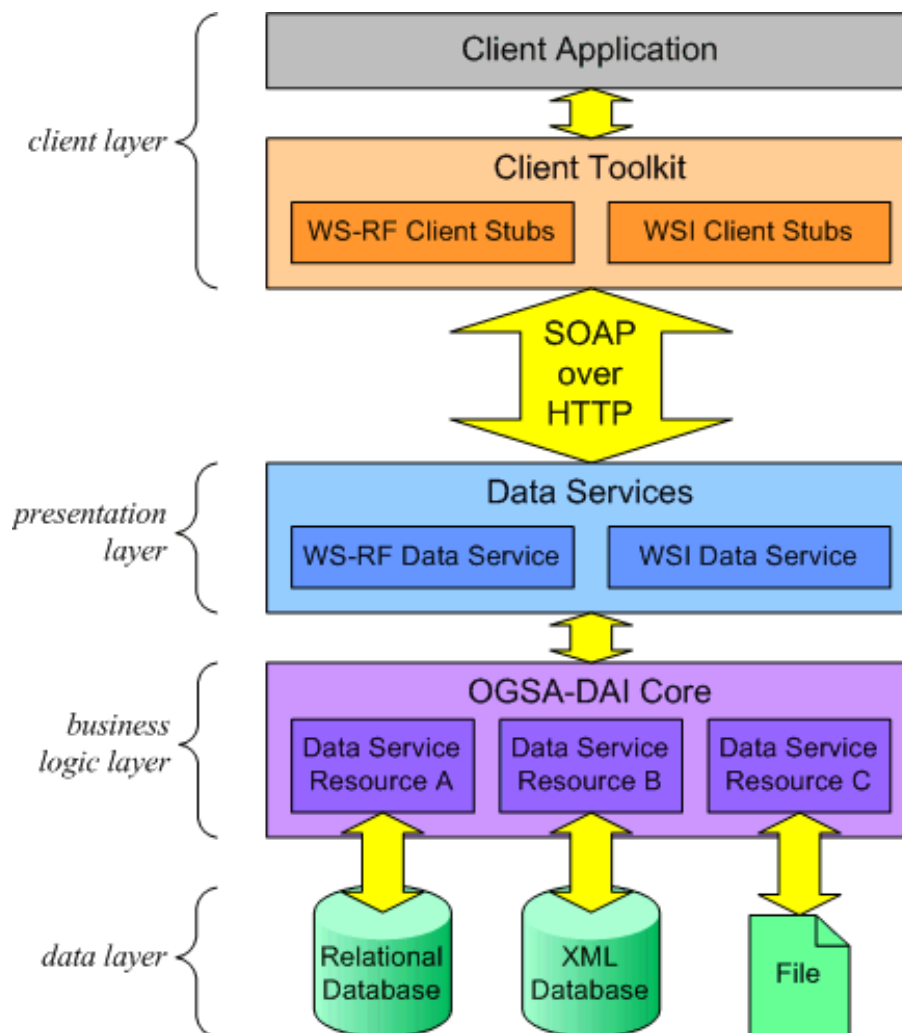


图 2.7: OGSA-DAI 体系结构图

(1) 数据层

OGSA-DAI可以访问各种数据源, 包括关系数据库、XML 数据库、文件。

(2)业务逻辑层

该层封装了OGSA-DAI的核心功能, 可以接受客户端的请求(如查询、修改等), 响应客户端的请求, 管理数据传输和发布, Session管理, 以及与数据源的连接、管理、交互等等。

(3)表示层

该层通过Web服务(WS)接口封装了OGSA-DAI向网络展示的所有服务, 主要有两种实现模式: 一种和WS-RF相匹配, 一种和WS-I相匹配。每种接口的实现, 都采用WSDL(WebServiceDescriptionLanguage, Web服务描述语言)和XMLSchema来描



述。

(4)客户层

OGSA-DAI支持与WSRF, WS-I兼容的客户端的访问,这主要依赖于服务器端表示层中提供的服务。通过客户端工具可以实现对底层数据源的检索、更新等操作;该框架中还提供数据层与业务逻辑层接口,可以调用JDBC驱动器,XMLDB驱动器等;表示层与业务逻辑层的接口负责在2层间的通信,支持对OGSA-DAI核心功能的调用。

2.4 国际虚拟天文台联盟(IVOA)及其相关规范

2.4.1 IVOA天文数据查询语言(ADQL) [17]

天文数据查询语言(ADQL)是被国际虚拟天文台联盟(IVOA)用来向VO数据服务(比如SkyNodes)提交数据查询的一种典型语言。为访问天文数据库, IVOA已经发展了多种标准协议,比如,针对天文图片数据的简单图片访问协议(SIAP)、针对光谱数据的简单光谱访问协议(SSAP)等。目前的ADQL 1.0.1版本仅仅支持星表数据查询,在将来的ADQL版本, IVOA将会把SIAP、SSAP以及ADQL统一整合起来。ADQL 1.0.1是基于标准的结构查询语言(SQL),主要是基于SQL92标准规范的。虚拟天文台的底层有大量的表格化数据集,并且许多这样的数据集存放在关系型数据库(RDBs)中的,所以使用SQL语句查询是一种非常方便的方法。但是ADQL是SQL的一个子集,IVOA根据天文数据的特点,对SQL92进行了必要的限制和扩展。

ADQL有两种表现形式:脚本形式(ADQL/s)和xml形式(ADQL/x)。这两者可以表现形式可以对等地相互转换。

ADQL和SQL的语法类似,他们主要不同点表现如下:

(1) 支持函数功能。包括计算大圆的距离、从六十进制到十进制的转换、单位换算等函数。例如:

```
Select HEALPIXID(a.ra, a.dec), a.ra, a.dec from photobjall a
```

(2) INTO 子句

INTO为专门支持将来的VOSpace的互操作性设置的。在SQL语句中,我们是用“SELECT INTO”去创建一个新表或者用“INSERT INTO”去向已经存在的表中添加数据。但是在ADQL中, INTO子句专门用来向VOSpace中添加或者创建文件或表的。例如:



Select g.* into VOS:/JHU/gal from galaxy g where g.redshift >3.5

(3) ADQL仅仅支持/* */的注释语法

(4) ADQL中所有的表名必须使用别名，别名的使用仅仅是标准SQL的一部分，但是在ADQL中别名的使用是强行的。例如：

Select * from table t (t就是table的一个别名)

(5) ADQL中增加了REGION关键字，REGION是在WHERE之句中用来限定查询范围的。在ADQL/s中的用法如下：

Region('CIRCLE J2000 19.5 -36.7 0.02')

在ADQL/x中，Region关键字表示为：RegionXML。

(6) 支持数学函数JDBC中的数学函数应该允许在ADQL中有效三角函数：

acos(x), asin(x), atan(x), atan2(x, y) 这里x和y是数字

cos(x), cot(x), sin(x), tan(x) 这里的x都代表半径。

数学函数：abs(x), ceiling(x), degrees(x), exp(x), floor(x), log(x), log10(x), mod(x, y), pi(), power(x, y), radians(x), sqrt(x), rand(), round(x, n), truncate(x, n) 这里x和y是数字，n代表整数。

(7) 交叉证认 (XMATCH) 功能

XMATCH是指两个或者多个天文星表之间的交叉证认。XMATCH是用在WHERE子句中，貌似一个函数，XMATCH有四个参数：前两个参数是需要交叉证认的表名，第三个参数是 κ 方匹配 σ 的值，最后一个参数N代表交叉模式。例如

SELECT o.objId, o.ra, o.r, o.type, t.objId FROM SDSS:PhotoPrimary o, TWOMASS:PhotoPrimary t WHERE XMATCH(o,t,3.5,1) AND Region('Circle J2000 181.3 -0.76 6.5') AND o.type=3

(8) 存档资格功能

(9) 支持XPATh，像支持SQL一样，ADQL支持XQuery，因为我们的有些数据使用XSD描述的。例如：

Select /Resource/Contact/Name from Resource where /Resource/Type like 'catalog'

(10) 支持TOP，返回子集记录数。例如Select top 10 g.* from galaxy g

(11) 支持单位，ADQL语句中支持所有的常数值带上单位。这是可选的做法，ADQL并不详细说明这些单位都是什么意思，他只是在语法上是允许的。例如：

Select g.* from galaxy g where g.gmag > 100 Jansky

(12) Table 名称支持特殊的字符 “[]”，当表的名称容易被解析错误时，需要用 “[]” 将表名括起来。例如：



Select a.* from [2df] a

一个ADQL/s 的例子, 如下:

```
SELECT a.objid, a.ra, a.dec FROM SDSSDR2:Photoprimary a WHERE Re-  
gion('CIRCLE J2000 181.3 -0.76 6.5')
```

将他转换成ADQL/x 形式如下:

```
<?xml version="1.0" encoding="utf-8"?>  
<Select xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xmlns="http://www.ivoa.net/xml/ADQL/v1.0">  
<SelectionList>  
<Item xsi:type="columnReferenceType" Table="a" Name="objid" />  
<Item xsi:type="columnReferenceType" Table="a" Name="ra" />  
<Item xsi:type="columnReferenceType" Table="a" Name="dec" />  
</SelectionList>  
<From>  
<Table xsi:type="archiveTableType" Archive="SDSSDR2"  
Name="Photoprimary" Alias="a" />  
</From>  
<Where>  
<Condition xsi:type="regionSearchType">  
<Region xmlns:q1="http://www.ivoa.net/xml/STC/STCRegion/v1.10"  
xsi:type="q1:circleType" unit="deg">  
<q1:Center>181.3 -0.76</q1:Center>  
<q1:Radius>6.5</q1:Radius>  
</Region>  
</Condition>  
</Where>  
</Select>
```

2.4.2 简单图像访问规范(SIA) [18]、简单光谱访问规范(SSA) [19]

SIA和SSA协议都是数据获取层(Data Access Layer)的协议。简单地说, SIA协议是在制定天区搜索并获取图像数据的协议, 而SSA则是在制定天区搜索、获取光谱的协议。SIA、SSA协议的工作过程都可以分为两个部分: 定位图像和获取图像。



定位图像中，它们都使用HTTP GET方法，访问CGI服务。客户端必须提供的三个参数是天区的中心位置（POS）、天区的尺寸（SIZE）和返回图像/光谱的格式（FORMAT）。POS是用逗号分割的、以度为单位的赤经和赤纬；SIZE则是用逗号分割的角距离；FORMAT则可以是多种格式，如image/fits, image/jpeg。而HTTP返回的则是VOTable格式的图像/光谱，在这个VOTable中，每行代表一个符合要求的图像。在此行的列VOX:Image_AccessReference中是这个图像的URL地址，客户端可以据此获取图像。按照SIA、SSA协议，服务器端传回的图片可能是本来就存在于服务器中的，也可由服务器临时生成。

2.4.3 FITS、VOTable等天文格式规范

FITS (Flexible Image Transport System) [20]是天文学界常用的数据格式，它专门为在不同平台之间交换数据而设计。1988年的国际天文学联合会（IAU）大会指定IAU的FITS工作组全权负责此格式的修订。IAU规定，今后对FITS标准的修改不得破坏前后一致性，也就是所谓的“once FITS, always FITS”一说。

FITS文件由文件头和数据组成。在文件头中存储有对该文件的描述，如观测时间、观测对象、拍照温度、曝光时间等信息，同时也可以文件头中注明观测时的视场、精度等，便于后期数据分析之用。文件头部分每行占80个字符，并以END结尾。按南京大学郑兴武教授所编《现代天体物理实验指导》一书中的说法，文件头共有36行，若不够，则再增加36行（或其倍数），未满部分以空格补之。每行的数据值一般为右对齐，“/”之后是注释内容。其具体表示格式是：

Keyword = Value /Comments

其中Value部分可以为空。较重要的几项有：BITPIX（指明图象位数）、NAXIS（指明图象维数）、NAXISn（代表第n维的像素数）。数据部分可以是图象或其他信息（如数据表），这些数据按头部所给出的描述组织。

这里给出FITS文件头的一个例子（引自AVIS FITSviewer所带的FITS样品）：

```
SIMPLE = T /  
BITPIX = 16 /  
Updated by AstroArt NAXIS = 2 /  
NAXIS1 = 228  
NAXIS2 = 242  
COMMENT COMMENT OBJECT = 'M57 Nebula'  
TELESCOP= 'Celestron C5'  
EXPOSURE= 120  
APOLY0 = 4.94607111393E+000
```




```
APOLY1 = 5.77546902777E-001
APOLY2 = -2.11265665729E-005
APOLY3 = -4.11369637119E-006
APOLY4 = 2.70776116332E-003
APOLY5 = -5.25005427724E-006
APOLY6 = 1.64728178524E-005
APOLY7 = -1.27203899586E-003
CTYPE1 = 'RA-TAN'
CTYPE2 = 'DEC-TAN'
CRVAL1 = 2.83389000000E+002
CRVAL2 = 3.30910000000E+001
CRPIX1 = 1.07521861153E+002
CRPIX2 = 1.12170084259E+002
CD001001= -1.21046310023E-003
CD001002= -2.35697440267E-004
CD002001= -3.00805952301E-004
CD002002= 9.43822939631E-004
CDELT1 = -1.24727909385E-003
CDELT2 = 9.72807804616E-004
CROTA1 = 0.00000000000E+000
CROTA2 = 1.39885366995E+001
AVISUMIN= 1.56000000000E+002
AVISUMAX= 1.05200000000E+003
AVISUTYP= -1.00000000000E+001
END
```

按IAU的规定, FITS文件格式有指定的几类, 如FITS、TABLE、IMAGE等, 具体可参考<http://fits.gsfc.nasa.gov/xtension.html>

VOTable [21]是XML格式的, 用来存储和传输天文星表数据及其元数据的标准。在VOTable标准中, 提供了对星表元数据的描述。元数据包括对星表自身的描述和对星表每一列的描述, 如表名、使用的坐标框架、数据类型、单位等等。同时它还引入了统一内容描述 (Unified Content Descriptors, UCD), 使程序可以理解每一列的具体天文含义。

天文文档常用格式还有ASCII、CSV等, 这里我们就不一一介绍了。

第三章 基于WSRF的数据访问服务(VO-DAS)

3.1 VO-DAS的需求分析

e-Science时代的天文数据有以下特点 [22]:

*天文数据具有异地、异构性。各国的观测资料，VO以前并没有形成一个国际标准。世界各国的天文资料保存和管理方式存在巨大差异，不同历史时期数据保存和管理方式也存在巨大差异。数据结构的混乱和数据存储的多样性就像语言不通一样，严重影响着天文学家对这些数据资料的使用效率。

*天文数据绝大部分是开放数据。国际上许多大型天文观测项目的观测数据都会及时在互联网上公布，这为数据共享提供了良好基础。

*天文数据很少有商业价值。这意味着相互之间不存在保密性的问题，可以将数据和结果进行自由共享，非常适合国际性的联合研究与试验。

*天文数据是海量的数据。例如美国正计划建造的“暗物质望远镜LSST [23]”，每天的观测数据高达30TB。这非常适合网格环境下海量数据处理和数据挖掘算法的研究。

*天文数据是真实而归档完好的数据，并提供在线访问服务。

*天文数据是高度复杂的数据。海量高度复杂的数据对数据处理、数据挖掘、可视化等研究提出了挑战。

基于以上的数据特点，在国际虚拟天文台联盟（IVOA）的统一协调下，各成员国一直着力于开发统一访问这些异地、异构数据的天文数据访问系统，这套系统应该具备以下特点：

- 能够访问异地、异构天文数据库；
- 支持自动发现数据资源；
- 支持特定空间区域的数据访问；
- 支持海量数据访问；
- 支持天文数据的交叉认证；
- 支持星表数据，图像数据和光谱数据的访问；
- 具有标准的数据访问接口；
- 具有良好的互操作性，可以和任何符合IVOA规范的VO应用程序互联。

中国虚拟天文台数据访问系统（VO-DAS）正是在这种需求下提出的。



3.2 VO-DAS的框架 [24]

3.2.1 VO-DAS五种访问接口

VO-DAS表现为一个基于WSRF的网格服务,因此它的接口通过一个WSDL文件就可以完全描述。我们将VO-DAS的接口分成五个类型:资源元数据接口(Resource Metadata Interface, RMI)、数据查询接口(Data Query Interface, DQI)、数据分析接口(Data Analysis Interface, DAI)、数据分发接口(Data Delivery Interface, DDI)和操作管理接口(Management Interface, MI)。

RMI为VO-DAS的客户端提供一套获得VO-DAS所发现的所有数据资源以及它们的详尽描述的方法。借助这套接口,客户端可以清楚知道数据查询会在哪些数据资源、哪些DataNode上进行,减少了数据查询的盲目性,有助于数据用户有目的有选择的进行数据访问,提高数据访问效率。

DQI是将ADQL的查询语言直接传送给VO-DAS的接口。数据查询分成两种情况:同步查询(Synchronous Query)和异步查询(Asynchronous Query)。同步查询的情况下,客户端发送查询请求以后等待结果数据通过网格服务的SOAP接口直接传递回来;而异步查询则要求客户端在提供ADQL查询语句的同时提供一个数据存放的目标URI。当查询开始后,客户端程序不需要等待返回结果,结果数据会自动上传到指定的目标URI的位置上。异步查询是专门为复杂的长时间查询设计的接口。

DAI是一个扩展功能接口,它将提供一个JDL描述的数据分析脚本,VO-DAS本身并不解析JDL脚本,而是调用它在网络中发现的CompuCell网格服务,把数据结果的URI和JDL脚本传递给一个CompuCell服务,由后者完成JDL描述的数据分析工作。有了DAI接口,VO-DAS就可以和虚拟天文台的数据挖掘系统配合完成数据访问和分析。

DDI是一套向VO-DAS指示如何传输数据查询结果的接口。

MI提供了session管理, ping等方法。Session用来管理异步查询和跟踪查询操作的状态。ping检查VO-DAS是否还能正常访问。

3.2.2 VO-DAS两大核心模块

VO-DAS的总体设计如图3.1所示。全系统分成两个部分:VO-DAS Center和DataNode。

3.2.2.1 VO-DAS Center

图3.1的上部框图是VO-DAS部分的设计示意图。VO-DAS框图的最上方是面向客户端的WSRF服务接口。VO-DAS的核心是Task Queue和ADQL Parser。

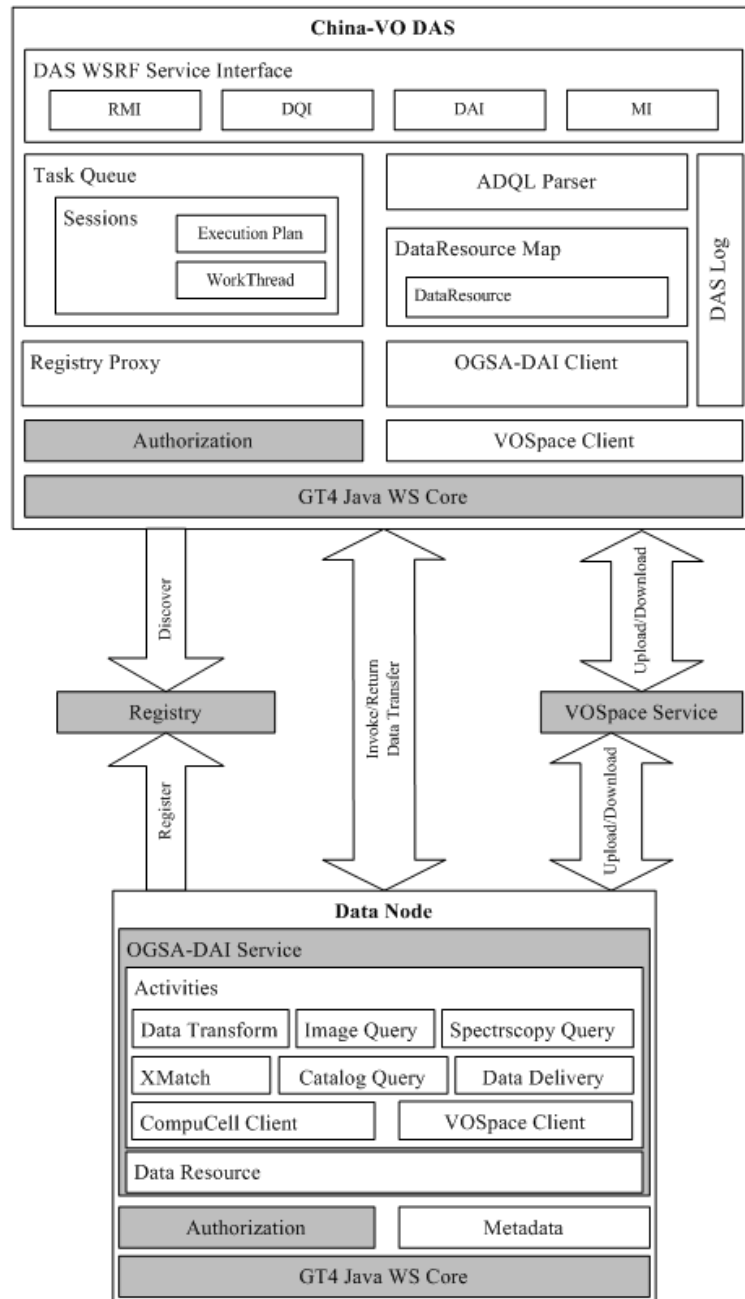


图 3.1: VO-DAS设计图

Task Queue负责调度和管理任务。每个来自客户端的查询请求都是一个任务。任务是使用唯一的session进行管理的。Session中保存任务的描述（ADQL查询语句，执行计划等）和输出结果（在同步查询时是结果数据集，在异步查询时是结果数据集的URI），维护一个工作线程（work thread）。在系统容量许可的情况下，Task Queue会按照先来先处理的原则调度session的工作线程。同时，Task Queue会监控



那些正在执行的工作线程的状态，当客户端发出运行状态查询请求的时候，它会将对应的任务的运行状态反馈回去。

当客户端发来一个查询请求的时候，它的ADQL查询语句被Task Queue送给ADQL Parser来进行解析和产生执行计划。考虑到ADQL的规范并没有成熟，我们在设计的时候有意将这个模块设计的相对独立，这有助于将来在ADQL规范确立以后的系统升级。ADQL Parser会通过查询DataResource Map来确定ADQL语句中包含的表是否存在以及存在哪些DataNode上。如果一个表有多个拷贝在不同的DataNode上，DataResource Map会帮助ADQL Parser决定使用哪个DataNode上的拷贝。ADQL Parser的最终输出是一个执行计划（Execution Plan）对象，它会保存到相关任务的session中等待执行。

3.2.2.2 DataNode

DataNode是VO-DAS系统中提供数据资源的服务，它是建立在OGSA-DAI WSRF中间件的基础上的网格服务。OGSA-DAI很好的封装了各种不同类型的数据库，统一了它们的访问接口。不仅是数据库，OGSA-DAI还支持将一个文件集作为数据库进行访问。OGSA-DAI的这些功能正好可以将不同结构不同组织和管理方式下的各种天文数据封装成为具有统一访问接口的数据资源，提供给虚拟天文台的各种数据分析工具使用。

但是，OGSA-DAI毕竟不是专门为虚拟天文台而开发的，一些天文学的特定的数据使用方式，例如天体在多个波段的交叉证认，还需要在OGSA-DAI的基础上进行扩展才可以实现。所幸的是，OGSA-DAI提供了一种类似于插件的Activity功能，使用者可以自己编写Activity程序让OGSA-DAI执行一些特定的查询和数据处理工作。因此，我们可以通过这个Activity功能实现在OGSA-DAI上的天体交叉证认，天文数据格式转换等特定工作。

DataNode在安装和配置成功以后应当注册到一个虚拟天文台Registry服务中。VO-DAS使用AstroGrid项目提供的Registry服务。DataNode的资源注册元数据遵守IVOA的Resource Metadata规范。

3.2.3 天文数据查询语言（ADQL）

IVOA制订的天文数据查询语言（ADQL）虽然已经更新了几个版本，但仍在草稿阶段。为了能够和以后的ADQL正式规范平稳衔接，VO-DAS采用了ADQL 1.01版本作为标准查询语言。ADQL是根据SQL92改进而来的针对天文数据访问特点而设计的数据库查询语言。例如：

```
SELECT p.ra, p.dec, p.r, t.j  
FROM sdss:photoprimary p, twomass:twomass_psc t
```



WHERE XMATCH(p,t,1) AND REGION('circle 157.5 55.1 1') AND p.r<22.5 AND t.j<14.5

这条ADQL语句的语法基本上和SQL相同,但有一定扩展。它查询SDSS的photoprimary星表和2MASS [26]的twomass_psc星表,其中SDSS是Sloan数字巡天观测项目 [25]的数据资源名称,2MASS是2微米全天巡天项目 [26]的数据资源名称。SELECT分句列出查询的列名,分别为photoprimary星表的赤经(ra)、赤纬(dec)、r波段的星等(r)和twomass_psc星表的J波段星等(j)。WHERE分句的第一个条件XMATCH表示这两个星表的天体做交叉证认(参与交叉证认的表为p和t,交叉证认的误差半径为1角秒)。这是一个ADQL的扩展函数。第二个条件REGION也是ADQL的扩展函数,表示查询和交叉证认都是在一个小的圆形天区(圆点在赤经157.5度,赤纬55.1度,半径1度)内完成的。第三和第四个条件语法同SQL完全一样。

通过上述例子我们看到,使用ADQL既可以描述对特定天空区域的查询也可以描述交叉证认,同时它独立于星表的物理存储位置和存储方式。

我们在VO-DAS中对ADQL的语义作了扩展,允许FROM分句中的表的类型是星表(等同于数据库中的表)、图像库(等同于文件集合)和光谱库(等同于文件集合)。当FROM分句列出的是星表的时候,整个ADQL的使用就是按照IVOA的标准进行;当FROM是图像和光谱的时候,SELECT分句的列名实际是图像和光谱的参数名,WHERE分句的条件则是对要查询的图像和光谱文件的参数进行的约束。这样,语法上ADQL没有任何改变,但是通过改变语义我们便让ADQL支持天文图像数据和光谱数据的查询了。

当ADQL查询语句中涉及到多个DataNode之上的表的时候,需要对异地异构数据库进行联合查询。对ADQL的分析会在VO-DAS之内进行,客户端的用户在编写这样的ADQL语句的时候不必考虑数据库的物理位置和类型。

3.2.4 数据查询

数据查询工作流有三种主要的情况:同步查询工作流、异步查询工作流和异地异构数据库联合查询的工作流。同步查询工作流是指客户端向VO-DAS发出请求后等待查询结果的返回;异步查询是指客户端在发出查询请求后立即返回,不等待查询结果,结果会以文件形式保存在一个URI处;联合查询异地异构的数据库是VO-DAS最复杂的功能,它将设计到数据节点间更加最方便的方式进行数据传输(三种查询的实现请参见下一章)。



3.2.5 数据分析

VO-DAS的数据分析主要是通过数据分析接口(DAI)将JDL以及CompuCell接入VO-DAS中,实现对数据资源的计算和分析(详细参见下章)。

3.2.6 数据分发

对于异步查询模式,数据查询完成以后会按照指定的数据格式将查询结果保存到指定的URI。这个过程就是数据的分发。VO-DAS支持多种数据分发的网络协议:HTTP、FTP以及GridFTP等。

3.2.7 服务的信息注册与信息发现

VO-DAS的资源注册和发现模块主要借用英国AstroGrid的相应模块。VO的核心思想就是共享,而且要做到全球共享。将来各国的VO都将形成一个联合体,所以借助英国AstroGrid的Registry模块,从长远看也是可行的。

3.2.8 数据格式

天文数据常使用的数据格式有FITS, ASCII, HTML, VOTable等。这些数据格式的转换可以通过在每个DataNode中添加格式转换Activity组件的方法实现。

3.2.9 VO-DAS的运行环境

DAS是一个独立的WSRF服务,和一个OGSA-DAI服务共同安装在GT4 Java WS Core之上。

`$DAS_HOME=$CATALINA/webapps/wsrf/WEB-INF/etc/das`是DAS的根目录

DAS的jar包保存在`$CATALINA_HOME/webapps/wsrf/lib`目录下

DAS的DAI服务安装在

`$OGSADAI_HOME=$CATALINA_HOME/webapps/wsrf/WEB-INF/etc/ogsadai`

目录下`das.xml`是它的配置文件,包括了所有的系统静态设置参数

3.2.10 VO-DAS的性能

- 1、查询:一次能够查询多于50万条数据
- 2、分析:一次可以分析多于500万条数据
- 3、交叉认证:一次可以做多于50万条的交叉认证
- 4、Session的生命周期:最长可以容许多于168小时



- 5、最少支持2个session同时工作，应允许多于100个session等待
- 6、Data Node数目大于5000
- 7、Algorithm数目大于100
- 8、数据传输率：接近网络最大值

第四章 VO-DAS任务调度研究与实现

4.1 多任务调度的几种可能方案

任务调度可以基于三种设计方案：

(1) 基于Web Service。SkyNode是基于Web Service设计的，支持异地异构的多波段天文数据的交叉认证。但是由于Web Service的限制，SkyNode仅仅能够用来对小数据集进行操作，无法完成大数据集的访问和交叉认证。同时Web Service是无状态的，我们只能借助其他方式，比如利用Tomcat容器来维护状态信息，所以Web Service无法担任异步操作的要求。

(2) 基于OGSI规范。该规范通过扩展Web服务定义语言WSDL和XML Schema的使用，封装资源的状态，将具有状态的资源建模为Web服务方式，来解决具有状态属性的Web服务的问题。但这方式有以下缺点：过分强调网格服务和Web服务的差别，导致了两者之间不能更好地融合在一起；Web服务和XML工具不能良好工作，可能会造成移植性差的问题；有些Web服务的实现不能适应网格服务的动态创建和销毁等等问题。如果VO-DAS基于OGSI会导致VO-DAS先天不足。

(3) 基于WSRF。Web服务资源框架WSRF采用了与OGSI完全不同的定义：资源是有状态的，服务是无状态的，很好地解决OGSI和Web服务之间存在的矛盾。WSRF的规范是针对OGSI规范的主要接口和操作而定义的，它保留了OGSI中规定的所有基本功能，只是改变了某些语法，并且使用了不同的术语进行表达；同时，使用WSDL 1.1定义OGSI中的各项能力，充分兼容现有的Web服务。基于WSRF的VO-DAS，同时融合OGSA-DAI及WS-Notification，可以有效地完成对大数据量的操作、异步操作等。

4.2 VO-DAS任务调度方案的设计

4.2.1 VO-DAS的Session机制

大家已经知道，Web Service不带状态的，WSRF是主要描述带状态（Resource）的Web Service框架，即 $WS-R = WebService + Resource$ 。VO-DAS对外发布的各个接口都是以WS形式发布出来的，而各个操作的中间或最终状态信息，是通过Session来保存的，这里的Session对应于WSRF框架中Resource。Session与WebService的搭配使用，如图4.1

这是一个Factory/Instance模式，SessionHome专门用来管理所有Session的一个

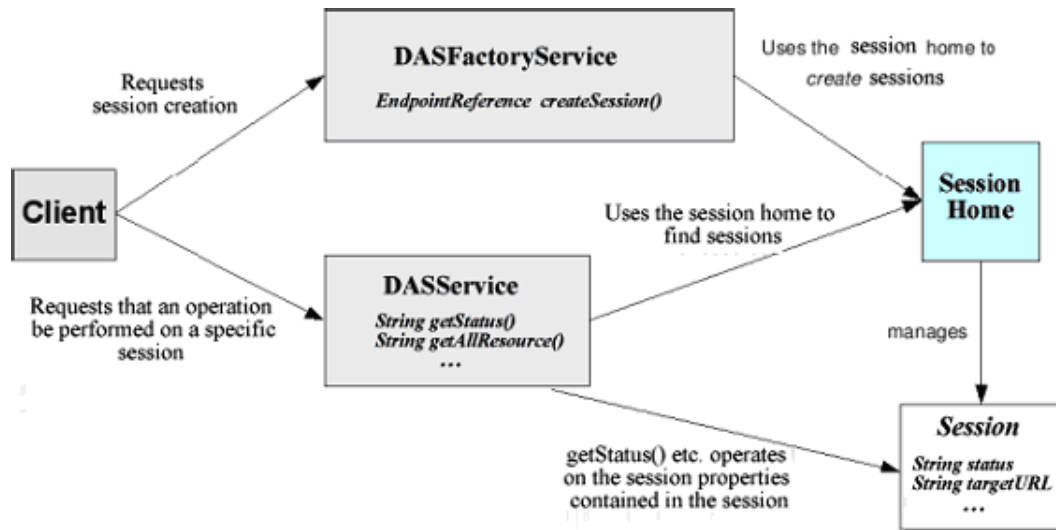


图 4.1: VO-DAS Session机制

类，Client端通过Factory Service调用Session Home的create()方法来创建Session；DASService是VO-DAS通过Web Service接口对外提供远程操作的一个类，Client可以远程调用DASService的相关方法，而每一个方法操作状态信息都可以通过与之配套的Session来保存。

4.2.2 VO-DAS的生命周期控制

VO-DAS的生命周期分为三类：全局对象的生命周期、非全局对象的生命周期和Session的生命周期。全局对象主要包括：LoopThread、WorkThreadPool、WaitingQueue、DataResourceMap等；非全局对象主要包括：WorkThread；Session主要包括：ExecPlan、status、targetURL等。我们这里主要讨论异步操作的Session的生命周期的控制（因为对于同步操作的Session 没有意义，我们不需要考虑）。

像LoopThread这样的全局对象的生命周期从VO-DAS启动到VO-DAS的结束。我们不需要过多考虑。

当LoopThead监控线程发现有排队的任务同时检测到工作线程池中有可用的工作线程时，非全局对象工作线程WorkThread才能生成，一直到这个线程将任务完成后，其自动结束，并有系统自动收回。WorkThread的生命周期主要由系统控制，我们也不需要过多考虑。

当Client端对VO-DAS有新的操作任务之前，VO-DAS将首先为这次操作生成一个利用SessionID标示Session，这是Session的生命的开始，但是Session的结束情况就比较复杂了，主要分为以下几种情况：



1. 每一个Session都有一个预设生命期限TimeOut。每一个Session在由Session-Home创建的时候, VO-DAS给它预先设置了一个生命期限, 比如不能超过6天(这个期限值可以在VO-DAS系统的配置文件中修改)。如果执行任务可以在预设生命期限内完成, 那么用户必须要在预设期限内提前相关数据结果(以后VO-DAS支持用户注册时, 可以将执行完后的状态信息永久保存在数据库中); 如果失效的Session“尸体”没有及时处理, 那么这种情况下, 失效Session在到达预设生命期限时, 自动由系统销毁。

2. 如果VO-DAS在执行某次任务的过程中, 程序出错抛出异常, 那么这次操作对应的Session也就失效。在这种情况下, VO-DAS在异常处理时利用Immediate Destruction方法将Session立即销毁, 同时将status置为ERROR。如果不立即处理这些Session“尸体”, VO-DAS可能会因资源耗尽而被“拖死”。

3. 执行任务尚未完成时, Session已经达到了预设生命期限TimeOut, 这时VO-DAS需要利用Scheduled Destruction方式管理生命周期。即, 通过周期性更新“租用时间”: 如果TimeOut到期, 发现status尚未等于Completed, 那么就延长TimeOut的值, 一直到执行完成。

4. 客户端主动销毁某次执行任务。这种情况, VO-DAS需要发布一个destroy()方法接口, 供客户端主动调用。其实, 这种情况同样是利用了Session的Immediate Destruction方法

4.2.3 VO-DAS的资源销毁

对于服务资源的销毁操作, 提供了两种方式: 一种是立即销毁(Immediate Destruction), 一种是调度销毁(Scheduled Destruction)。

1. 立即销毁(Immediate Destruction)

当客户端不再有任何请求时, 或客户端主动发出一个立即销毁资源的请求时, 服务器将会对不再使用的服务资源进行立即销毁处理, 以便收回资源占用的系统资源。

在VO-DAS调度中比较典型的场景: 客户要删除某次任务操作, 向服务器端发出请求, 或者是服务器端发现某次任务在执行过程中, 抛出异常, 导致任务无法继续进行, 即系统Status被置为ERROR时, 系统会立即销毁这次任务对应的Web服务资源。

如果Web服务资源没有响应销毁请求, 则必须将返回异常信息:

(1) ResourceUnknownFault(服务资源标示无法识别): 在销毁请求文档中的服务资源标示不能被服务器端识别。



(2)ResourceDotDestroyedFault(服务资源不能被销毁): 服务资源可能由于某种原因不能被销毁。

2. 调度销毁(Scheduled Destruction)

一种基于时间“租用”的方式来管理Web服务资源,在这种情况下,Web服务资源具有一个与之相关联的生命终止时间,这个时间定义了该Web服务资源预期被销毁的时刻,这个时刻之前Web服务资源是有效的,超过这个时刻,系统将自动销毁该Web服务资源。Web服务资源生命终止时刻可以通过检查TerminationTime资源特性来获取,也可以通过SetTerminationTime方法还改变终止时刻。

在VO-DAS调度中比较典型的场景,每个Session被创建时,都设定了最大生命周期,当系统发现在这个生命期内,对应的操作仍没有完成。这是,系统将通过SetTerminationTime方法,延长Session的生命周期。

Web服务资源为了支持调度销毁,Web服务资源必须包含如下的属性定义:

(1) CurrentTime(当前时刻),标示当前时刻。

(2) TerminationTime(销毁时刻),设定预期销毁时刻。

如果Web服务资源不能在预期时刻成功被销毁,则将返回异常信息:

(1)ResourceUnknownFault(服务资源标示无法识别): 在销毁请求文档中的服务资源标示不能被服务器端识别。

(2)UnableToSetTerminationTimeFault(不能设置销毁时刻错误): 服务资源可能由于某种原因不能设置销毁时刻。

(3)TerminationTimeChangeRejectedFault(更改销毁时刻被拒绝错误)。

4.2.4 数据查询设计

数据查询工作流有三种主要的情况: 同步查询工作流、异步查询工作流和异地异构数据库联合查询的工作流。

4.2.4.1 异地异构数据的同步查询调度方法

同步查询工作流是指客户端向VO-DAS发出请求后等待查询结果的返回。这是其中最简单的一个(如图4.2所示)。同步查询工作流的执行步骤如下:

A. 客户端程序向VO-DAS提交ADQL查询请求;

B. VO-DAS解析ADQL, 检查ADQL中涉及的数据资源所在的DataNode是否存在;

C. 如果DataNode存在, 则生成DataNode的执行计划对象, 并连接DataNode, 执行这个计划;

D. 从DataNode获得执行结果, 将执行结果返回给客户端。

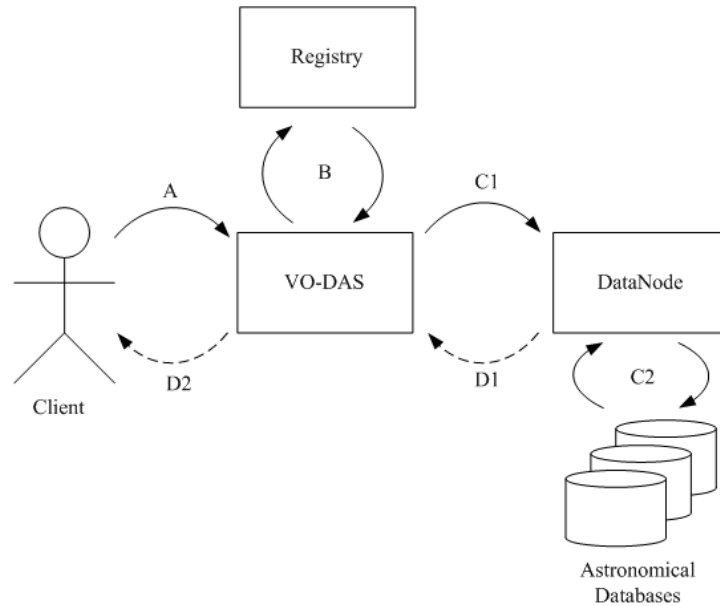


图 4.2: 同步查询工作流

4.2.4.2 异地异构数据的异步查询的调度方法

异步查询是指客户端在发出查询请求后立即返回，不等待查询结果，结果会以文件形式保存在一个URI处（如图4.3所示）。异步查询工作流的执行步骤如下：

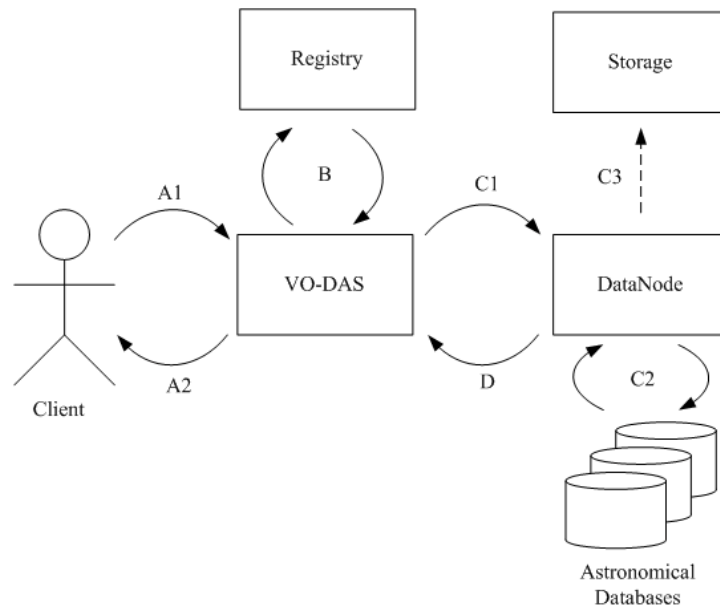


图 4.3: 异步查询工作流



A.客户端将ADQL查询语句和保存结果的URI传递给VO-DAS，然后立即返回，VO-DAS为这次查询建立session；

B.VO-DAS解析ADQL，检查ADQL中涉及的数据资源所在的DataNode是否存在；

C.如果DataNode存在，则生成DataNode的执行计划对象，并连接DataNode，执行这个计划，查询的结果保存到指定的URI（图中4.3虚线部分）；

D.结果保存完毕，DataNode向VO-DAS通知结束，VO-DAS删除相应的session。

4.2.4.3 异地异构数据联合查询的调度方法

联合查询异地异构的数据库是VO-DAS最复杂的功能，它的工作流见图4.4，具体步骤如下：

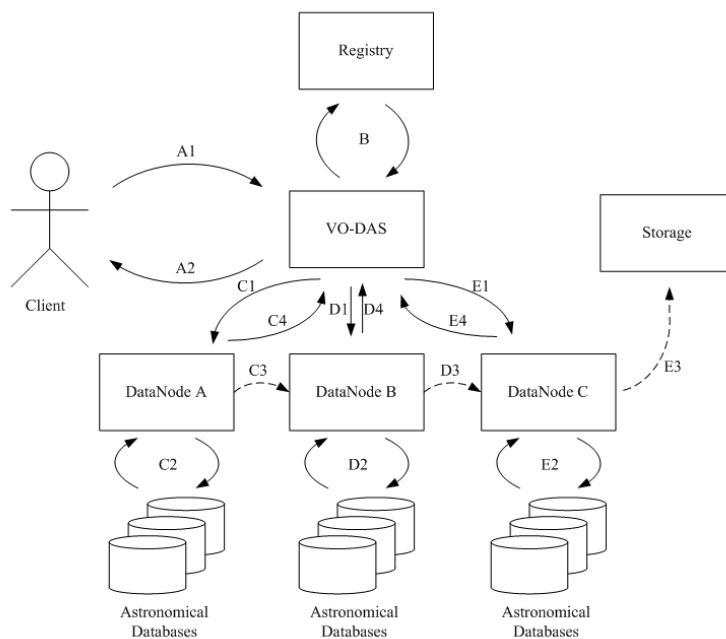


图 4.4: 联合查询工作流

A. 客户端将ADQL查询语句和保存结果的URI传递给VO-DAS，然后立即返回，VO-DAS为这次查询建立session；

B. VO-DAS解析ADQL，检查ADQL中涉及的数据资源的DataNode是否存在，如果存在则按照传输数据最少的原则为所涉及的DataNode排序并生成每个DataNode的执行计划对象（假设这里涉及三个DataNode，排出顺序为DataNode A, DataNode B, DataNode C）；



C. 向DataNode A发送执行计划, DataNode A完成查询以后将临时结果直接传递给DataNode B (图4.4中C3虚线表示数据传输方向), 数据传输后向VO-DAS报告DataNode A的查询结束;

D. VO-DAS向DataNode B发送执行计划, DataNode B会将来自DataNode A的中间结果保存在数据库临时表中并完成和这个临时表的联合查询, 联合查询的结果直接传递给DataNode C (图4.4中D3虚线表示数据传输方向), 数据传输后向VO-DAS报告DataNode B的查询结束;

E. 和D的过程相似, 只是最后的联合查询结果根据客户端指定的URI存放到一个外部存储空间中 (图4.4中E3虚线所示), 保存数据后会通知VO-DAS, 后者会删除相应的session。

4.2.5 数据的分析

VO-DAS的数据分析主要是通过数据分析接口 (DAI) 将JDL以及CompuCell扩展到VO-DAS中, 实现对数据资源的计算和分析。

4.2.5.1 JDL [27]

JDL是中国虚拟天文台开发的一种描述工作流程的语言(Job Description Language), 引入JDL的目的创建一个任务的WorkFlow。

JDL像ADQL一样, 有两种表现形式: 一种是通过脚本描述JDL, 即JDL/s; 一种是通过XML描述JDL, 即JDL/x。这两种形式可以相互转换, 前者主要面向用户的, 因为它方便书写, 实际在执行的过程中, 前者都转换成了后者, 因为后者便于程序识别。例如:

JDL/s形式:

```
project cc
job gettable
function t=main()
t=query( "select glon,glat, j_m, h_m, k_m from TwoMass
where glon>=270 and glon<271 and glat>-10 and glat<10" );
t=addcol(t, 5, "h-k", t("h_m")-t("k_m"));
t=addcol(t, 6, "j-h", t("j_m")-t("h_m"));
end
end
job cchist
function m=main()
```



```
t=jobresult(“gettable”);  
m=hist(t, “h-k”, “j-h”);  
end  
end  
end
```

等价转换成JDL/x形式:(请参见附录A)

4.2.5.2 CompuCell [27]

CompuCell是计算单元的缩写,在VO-DAS中CompuCell将表现为计算节点Algorithm Node(类似于数据查询情况下的数据节点DataNode),这些计算节点也会在注册服务中注册。当用户从VO-DAS的DAI接口提交一个JDL描述的数据分析脚本时,VO-DAS本身并不解析JDL脚本,而是调用它在网络中发现的CompuCell网格服务,把数据结果的URI(这里的URI可以是由VO-DAS完成的数据查询后,保存结果的远程的服务器的URI)和JDL脚本传递给一个CompuCell服务,由后者完成JDL描述的数据分析工作。

4.2.6 系统的状态监控

VO-DAS提供异步操作,这就意味着当用户在VO-DAS上提交任务后,用户就可以去干其他的工作,甚至可以将自己的电脑关机,所有的任务都会由VO-DAS来自动完成。对用户来讲,用户只需要VO-DAS处理任务的状态,当任务被执行完成后,前往目标路径targetURL处查看VO-DAS最终处理的结果。这就要求,VO-DAS必须要对用户提供一个获取状态的接口,即getStatus(),用户可以不时地调用getStatus()方法来查看任务的执行进度。

Status状态值,有以下几种情况:

1. PREPARING: 任务已经提交,VO-DAS正在为其生成执行计划ExecPlan。
2. WAITING: 执行计划已经完成,并且已经将其加入到排队队列WaitingQueue中。
3. PROCESSING: 任务正在执行。
4. COMPLETED: 任务已经正常完成,并将最终结果保存在用户指定的远程空间中。
5. DELETING: 任务正在被用户删除。
6. DELETED: 任务已经删除。
7. ERROR: 任务执行出错,并会抛出异常。



4.2.7 VO-DAS异常处理

异常几乎总是意味着某些事情不对劲了，或者说至少发生了某些不寻常的事情，我们不应该对程序发出的“求救”信号保持沉默或无动于衷。VO-DAS是一个相当复杂的系统，系统的层次关系相当复杂，如果系统在出现异常，但不抛出，用户根本不知道系统哪里出现了问题，也就无法“对症下药”，所以，系统必须在任何可能出现异常的地方，从最低层一级一级往上抛出异常，一直到最高层。

程序处理异常通常有以下几种情况：

1. 捕获异常但不做相应处理，这种情况我们通常调用`printStackTrace()`、`getMessage()`或者`getStackTrace()`等方法。这对我们调试程序时很有帮助，但是当程序调试结束后，我们就不应该再用这种处理方式。

2. 捕获异常但是不捕获具体异常，不管下级抛出多少异常、抛出什么样的异常，我们仅仅只用一个`catch`语句来捕获，最常见的情形是使用`catch(Exception ex)`。由于大多数异常都直接或者间接从`java.lang.Exception`派生，`catch(Exception ex)`就相当于说我们想要捕获所有的异常，这种捕获异常的方式就没有任何意义了，等于没有捕获。所以，这种情况一般也只能在程序调试阶段，为了简单使用。

3. 捕获到异常后不做相应处理，把异常处理重抛给上级处理。处理异常的代码在分析异常后，认为自己不能处理它，重新向上级抛出异常在VO-DAS中也是一种常用的处理方式。

4. 捕获到异常并作相应处理。这种情况在VO-DAS的调度程序中非常重要，比如VO-DAS在执行任务的时候，不管哪个环节出现异常，必须要报告上级，上级捕获到异常后，第一件事就是要将`status`置为`ERROR`，然后再做其他的处理，比如销毁这次操作的`Session`，释放资源。资源不释放是影响系统性能的潜在杀手。

5. 捕获到异常后将异常转换成另一种异常。对于比较低级的异常，用户不容易理解，这时我们可以将异常转换成应用级别的异常。

VO-DAS的部分异常列表(如表4.1)：

4.2.8 VO-DAS安全机制

VO-DAS将会基于GT4的安全机制GSI，对于客户端，我们将会对每一个用户设置一个账户（包括用户名、密码以及域名等），由于目前VO-DAS尚处于基本功能实现阶段，故安全机制暂时不做详细介绍。



表 4.1: VO-DAS的部分异常列表

异常名称	异常说明
CDASException	一般性异常
CDASDataNodeNotFoundException	数据资源结点没有找到
CDASDataInsertToTableException	数据插入表失败
CDASUploadDataFailedException	上传数据失败
CDASSessionFailedException	Session失败
CDASSessionNotFoundException	Session没有找到
CDASADQLErrorException	ADQL解析错误
CDASOutOfMemoryException	Data Node内存溢出
CDASDataNodeQueryException	Data Node查询错误
CDASMaxThreadReachedException	达到最大工作线程
CDASDataDeliverException	数据传输错误
CDASDataNodeOutOfMemoryException	DataNode服务内存不足
CDASDataURLNotFoundException	DataURL没有找到
CDASAlgorithmNotFoundException	算法运行库没有找到
CDASAlgorithmAccessException	算法运行库访问错误
CDASTimeoutException	DAS系统超时

4.3 VO-DAS任务调度的实现和测试

4.3.1 结构图（以三个DataNode为例）

以三个DataNode为例，实现三种查询，同时实现数据的交叉认证，最终将数据保存于VOSpace中。请参见图4.5

4.3.2 类图

请参见图4.6

4.3.3 序列图

4.3.3.1 系统的启动

请参见图4.7

4.3.3.2 系统的关闭

请参见图4.8



4.3.3.3 元数据查询

请参见图4.9

4.3.3.4 同步查询

请参见图4.10

4.3.3.5 异步查询

请参见图4.11

4.3.3.6 上传数据

请参见图4.12

4.3.3.7 数据分析

请参见图4.13

4.3.4 测试方案和测试结果

China-VO小组采用两个科学范例对VO-DAS原型进行了实验。实验的目的是验证以OGSA-DAI为基础的天文数据访问的可行性和性能。

第一个科学范例是采用Sloan数字巡天星表 [28]中的恒星星表作为数据源,通过研究这些恒星在空间分布密度的特点寻找银河系的卫星星系或球状星团 [29]。在该范例中,客户端程序会向VO-DAS发出数百个空间位置查询,每得到一个空间位置的数以千计的查询结果就对这些恒星的空間密度、亮度、颜色特征等进行分析,给出可视化结果供人工判别。该范例验证了VO-DAS以及OGSA-DAI在较强的访问密度下进行查询的稳定性以及大量数据传输的可行性。但是当访问密度非常高(对同一个约7000万条数据的大表同时有3个客户端连续发出请求)的时候,可能会出现偶发性的错误。发生偶发错误的原因尚不能确认,可能是我们使用的MySQL数据库的问题也可能是OGSA-DAI的自身问题。

第二个科学范例是对Sloan数字巡天星表的星系红移做出估计 [30],并使用估计的结果完成星系大尺度结构的可视化。在此范例中,客户端需要向VO-DAS以及DataNode请求超过2400万条星系位置和红移数据,并利用这些数据完成星系大尺度结构的绘图。实验发现,一次访问2400万条数据会受到实验计算机的内存限制(我们使用1.5GB内存的个人计算机进行实验),适当降低一次访问数据的数量可以让查询稳定进行。在我们实验的环境下,我们一次访问最多达到500万条数据。在



具体实践中,一次访问的允许最大访问数据量应该同服务器的内存大小以及每条数据的字节长度有关系。

两个科学范例的实验说明,采用OGSA-DAI为核心的VO-DAS系统对于支持天文学的数据访问是可行的。运行性能在目前阶段还仅仅是满足于可用。我们将在以后的工作中逐步提高系统的运行性能,使其成为不仅可用而且好用虚拟天文台应用系统。

4.4 和国外同类虚拟天文台系统的比较

选择OGSA-DAI可以有效封装异构天文数据库,采用我们设计的VO-DAS,可以在异地的DataNode之间实现异构数据库联合查询,进而实现天文学多波段数据的交叉证认。同时OGSA-DAI的Activity插件使得我们可以很好的封装对天文图象数据和光谱数据的访问,让这些访问看起来就像访问普通的数据库。借助AstroGrid开发的Registry服务,VO-DAS可以自动发现DataNode并对这个DataNode进行访问。实验表明,我们可以支持数百万条数据的直接访问。作为比较,NVO的OpenSkyQuery实验原型仅能够支持最多5000条数据。由于我们采用了IVOA的规范来约束外部接口,所以系统将获得良好的互操作性,可以和将来更多的虚拟天文台应用程序实现互联互通。

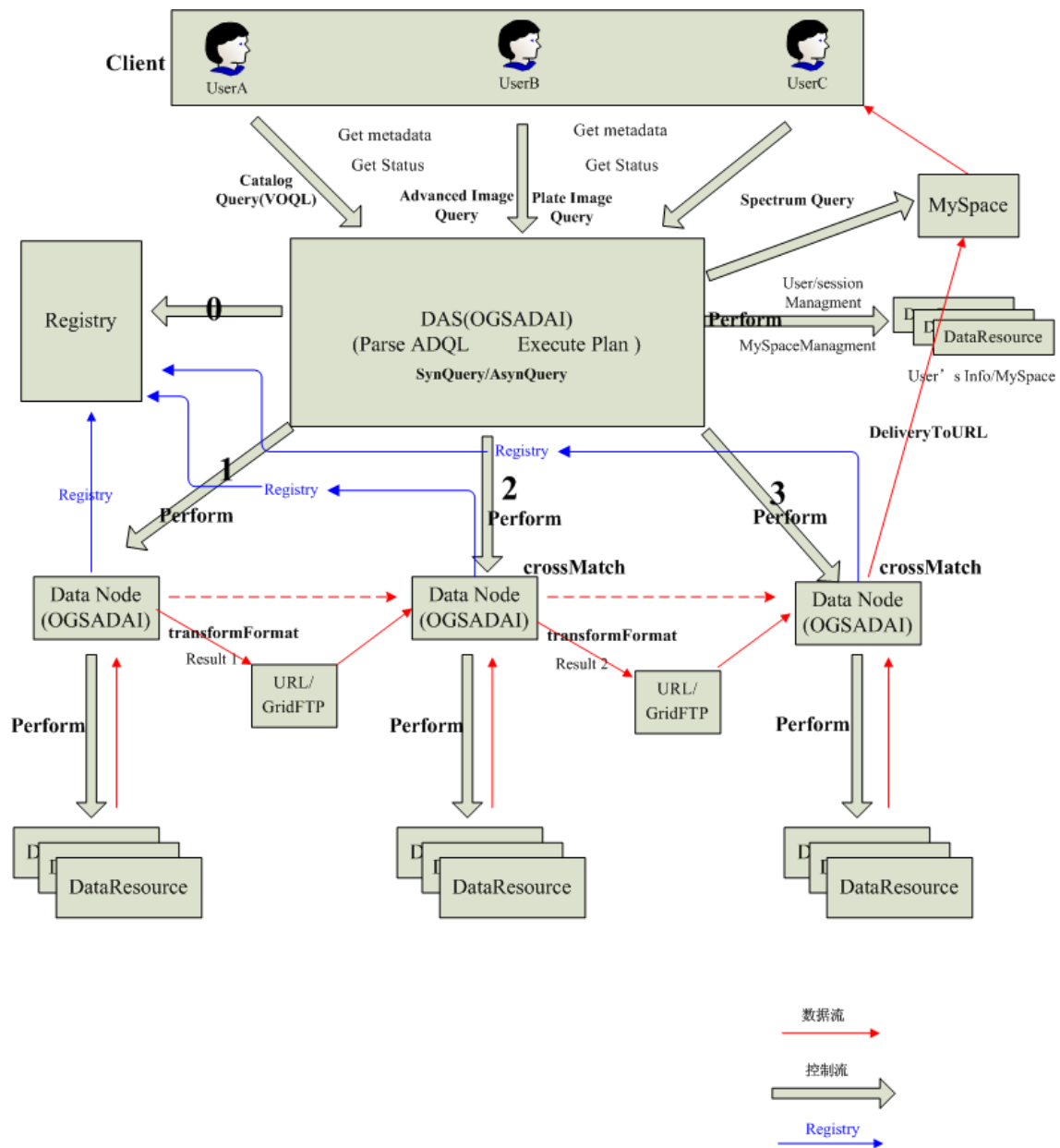


图 4.5: VO-DAS 结构图 (以三个DataNode为例)

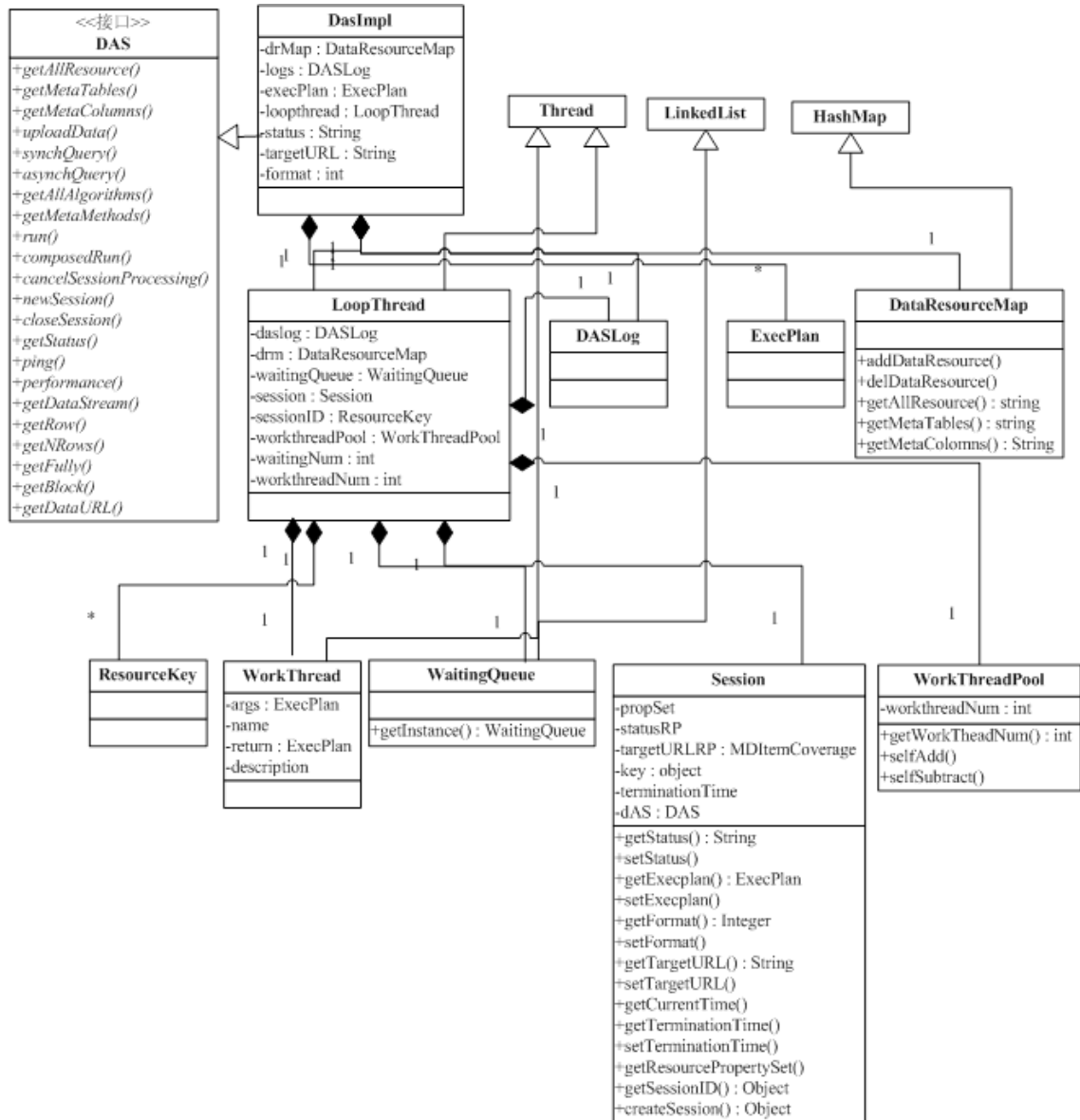


图 4.6: VO-DAS 类图

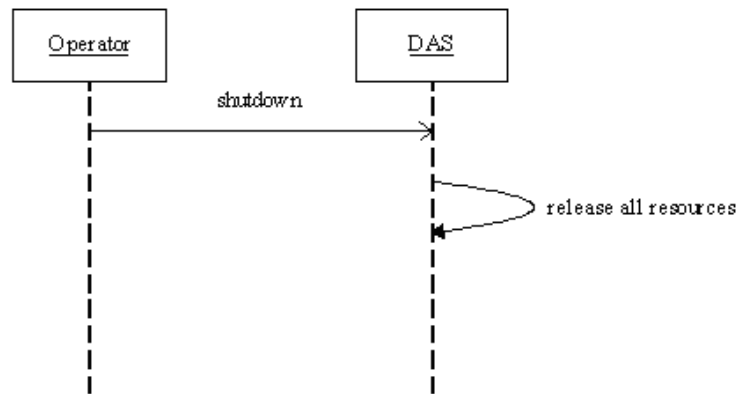


图 4.7: VO-DAS的启动

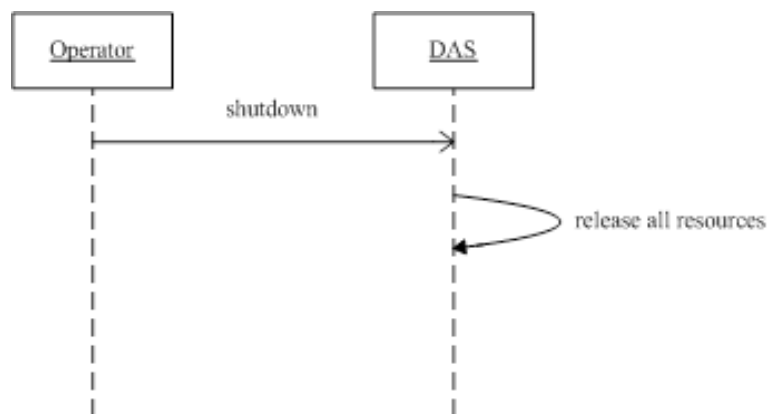


图 4.8: VO-DAS的关闭

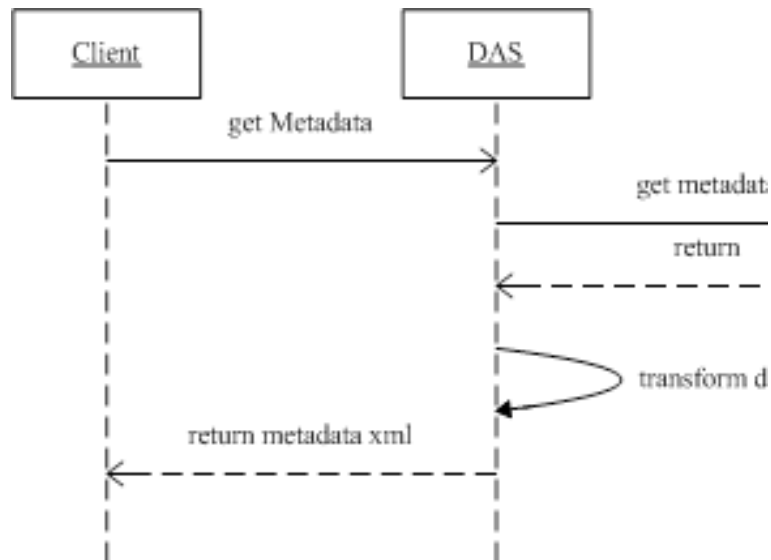


图 4.9: VO-DAS的元数据查询

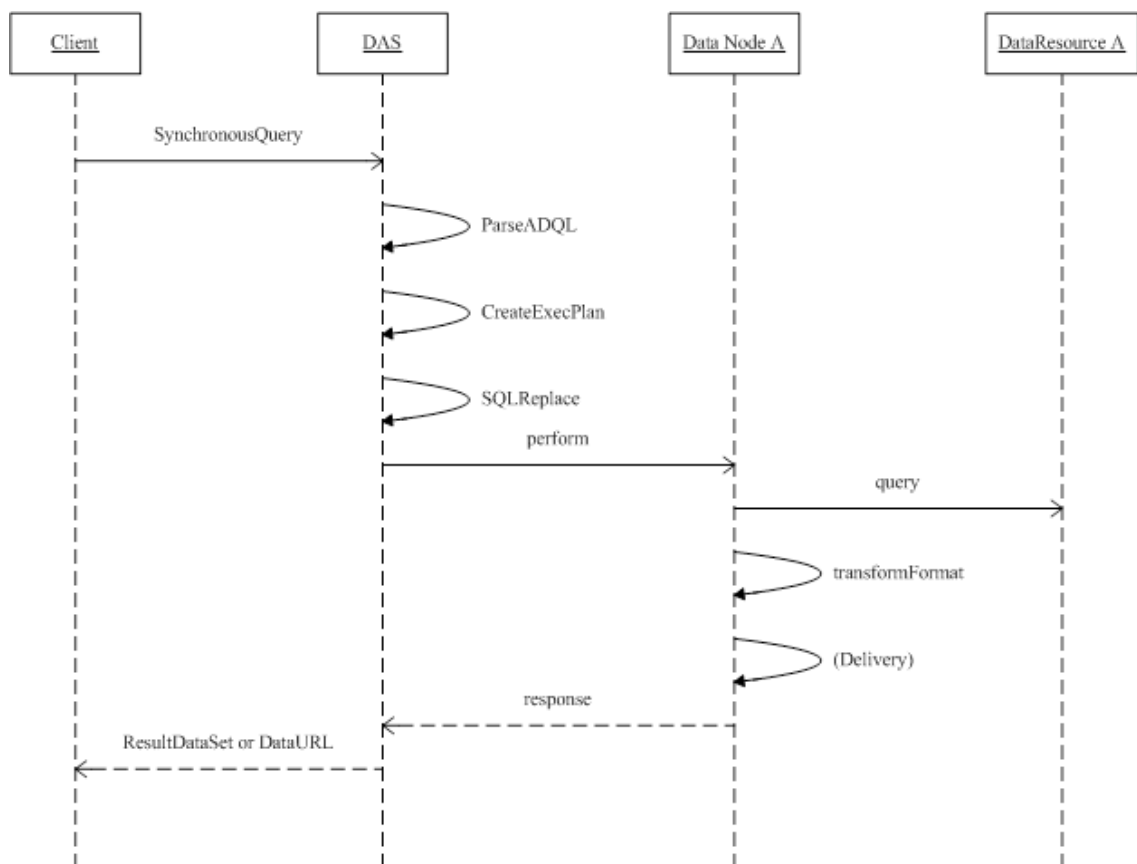


图 4.10: VO-DAS的同步查询

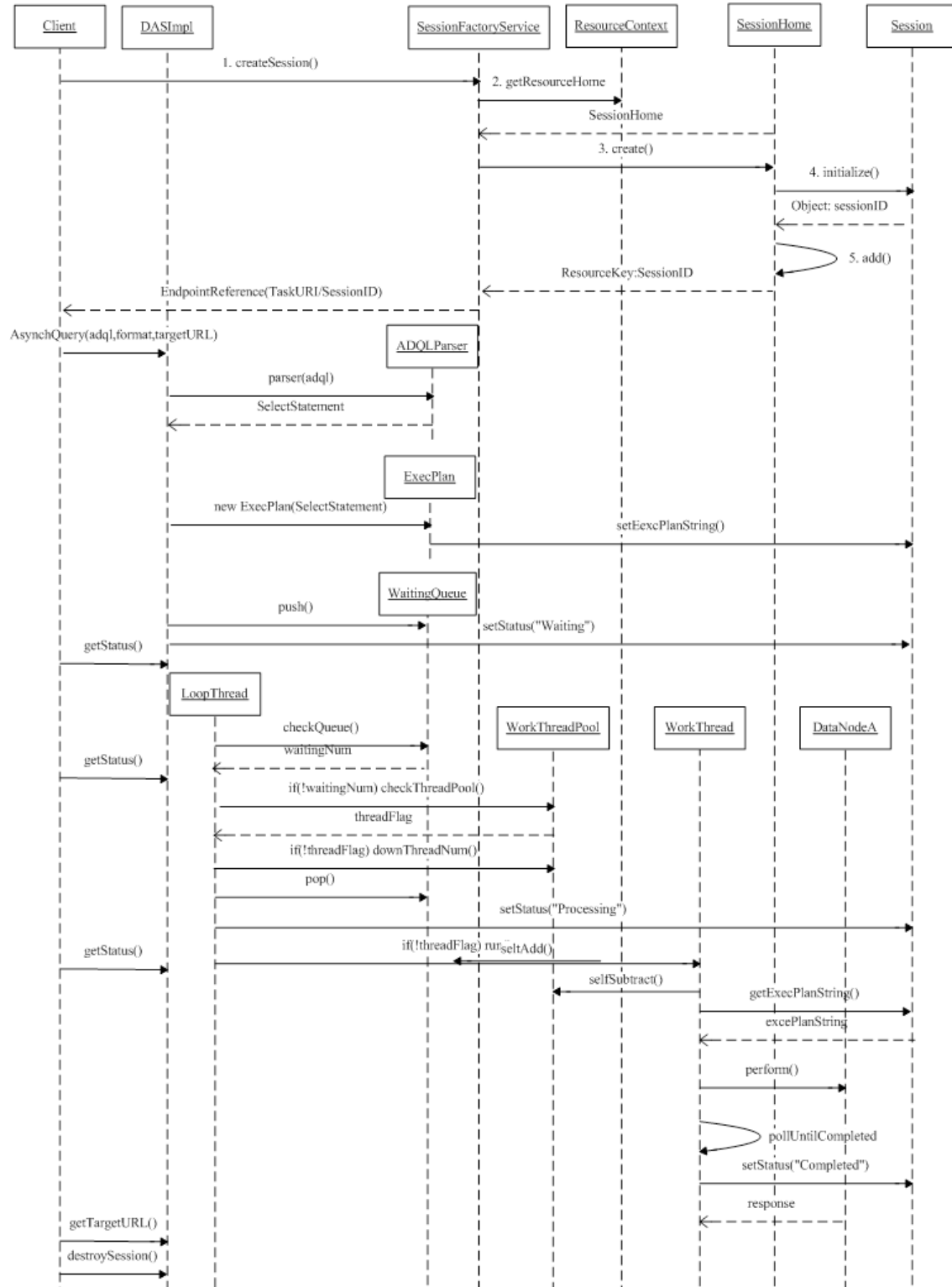


图 4.11: VO-DAS的异步查询

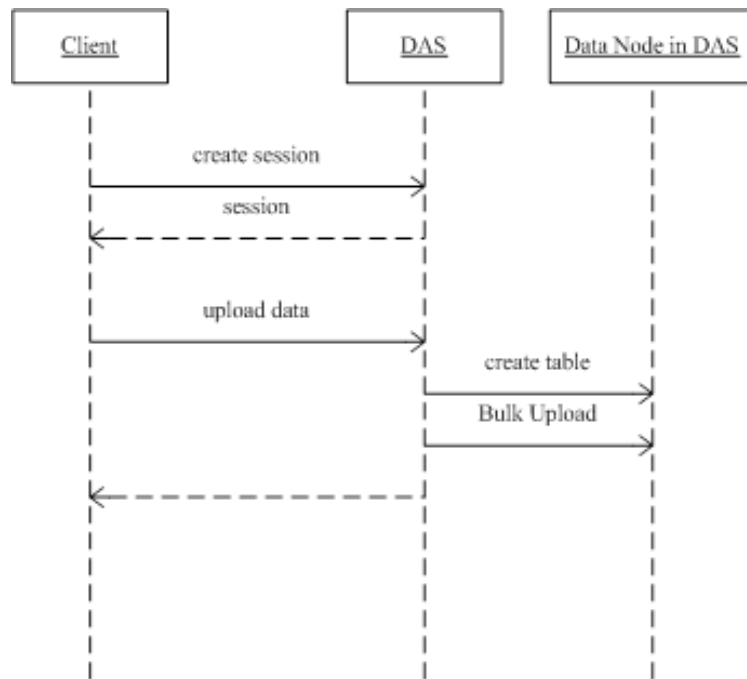


图 4.12: VO-DAS上传数据

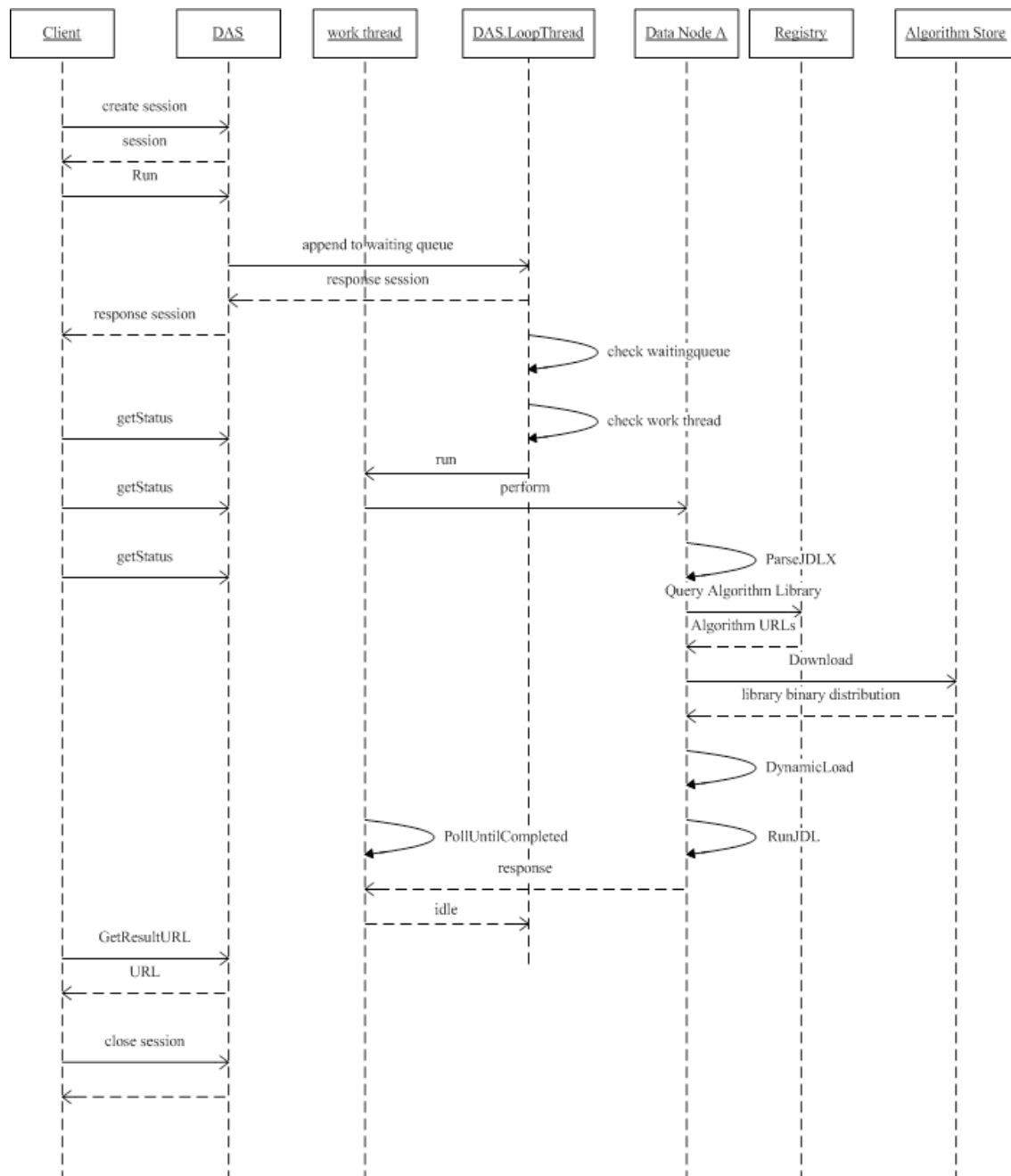


图 4.13: VO-DAS的数据分析

第五章 VO-DAS的功能扩展及其应用

5.1 基于VO-DAS平台可视化日食带的中心食带

在基于VO-DAS平台可视化日食带的中心食带案例中，我们通过OGSA-DAI的Activity把China-VO Ephemeris WS 计算平台封装成VO-DAS的一个计算节点，并且在该节点上实现一个比较简单的可视化应用程序，用户可以以VO-DAS的标准访问方式来进行星历的常规计算。

5.1.1 China-VO Ephemeris WS 计算平台 [31]

为克服复杂耗时的星历计算，基于兼备计算精度高、时间跨度长、内容丰富等优点的Swiss Ephemeris星历计算程序包，利用JNI和Web Service等新兴的技术，我们设计出了一套面向星历研究人员和天文程序开发人员的星历计算服务平台—China-VO Ephemeris WS。

5.1.1.1 Swiss Ephemeris [32]

Swiss Ephemeris是由Astrodienst基于美国宇航局JPL实验室发布的DE406星表拓展而来的一个高精度的星历表。它不是一个面向终端用户的产品，而是面向天文软件程序开发者的一个开源工具集。它给我们提供了如下一些计算功能：行星和恒星的计算；食、掩星、行星的升降等行星现象的计算；日期和时间转换功能；黄道十二宫的计算以及一些辅助功能。

Swiss Ephemeris比其他星历表有以下优点：

A.占用空间小：利用了复杂的压缩技术，对于6000年的行星数据只需5M磁盘空间，月亮数据需要13M空间。

B.跨度时间长：-5400年1月2日 5399年12月31日，比其他星历表的6000跨度扩展到10800年，实现了真正的万年历。

C.精度高：所有需要考虑的修正如相对论性偏差、光在太阳重力场中的弯曲等实际因素都以极高的精度加以考虑使整体转换精度达到0.001角秒。

D.可以利用三种不同的星历表（原始的JPL DE406数据、经过压缩处理后的Swiss Ephemeris数据（默认选项）、基于Moshier model DE406数据）进行计算，该软件包将根据所安装的数据文件自动切换到精度最高的星历表进行计算。

E.支持三种坐标系：赤道坐标系、笛卡尔坐标系（XYZ坐标系）、黄道坐标系。同时支持三种坐标系的相互转换。



F.内容丰富: 包含了所有已知的小行星甚至是假想天体的计算, 目前它包含了55000颗小行星的数据。

G.运算速度快: 在1000MHz的PIII处理器的测试机器上, 计算10000颗星的位置只需9秒钟。

基于以上优势, 我们选择Swiss Ephemeris作为平台的后台处理程序。Swiss Ephemeris支持VB, C等优秀语言, 但不支持Java, 为了Swiss Ephemeris支持Java, 这里我们用JNI技术, 将Swiss Ephemeris的C 程序包中各服务接口转换并封装成Java本地化调用接口。

5.1.1.2 JNI [33]技术概要

JAVA通过JNI (Java Native Interface) 调用本地异类 (C程序) 方法, 而本地方法是以库文件的形式存放的 (比如在WINDOWS平台上是DLL文件形式, 在UNIX机器上是SO文件形式)。通过调用本地的库文件 (即系统级的各接口方法), 实现JAVA程序和本地异类程序的紧密联系。在这里我们把Swiss Ephemeris作为本地方法, 利用JNI技术将Swiss Ephemeris的C程序包, 转化成Java能够调用的库文件。

JNI技术的具体书写步骤:

- A. 编写带有native声明的方法的java类
- B. 利用命令javah -jni 将java类转换成扩展名为.h的头文件
- C. 使用C/C++实现本地方法
- D. 将C/C++编写的文件生成动态连接库

5.1.1.3 平台的结构设计

平台主要包括五大功能 (如图5.1): 行星和恒星的计算 (行星、恒星、月亮、小行星以及假象天体)、日期和时间的转换、黄道十二宫的计算、行星现象 (食、掩星、行星的升降等)、其他辅助功能 (坐标转换等)。其中每种功能都是由多个函数模块来完成, 比如食现象包括: 给定儒略日计算下一次发生日食的时间swe_sol_eclipse_when_glob(), 给定儒略日、地球的经纬度以及海拔高度来计算日食的属性swe_sol_eclipse_how (), 计算下一次月食发生的时间swe_lun_eclipse_when()等等功能 (更加详细的功能列表可以参见Programming interface to the Swiss Ephemeris文档 [34])。为了方便发布Web Services, 我们利用JNI技术将Swiss Ephemeris的各接口发布成Java接口, 我们称之为China-VO Ephemeris程序接口, 然后利用Axis将China-VO Ephemeris各接口发布成Web Services, 这样程序员就可以非常方便地调用这些Web Services, 实现自己的开发。

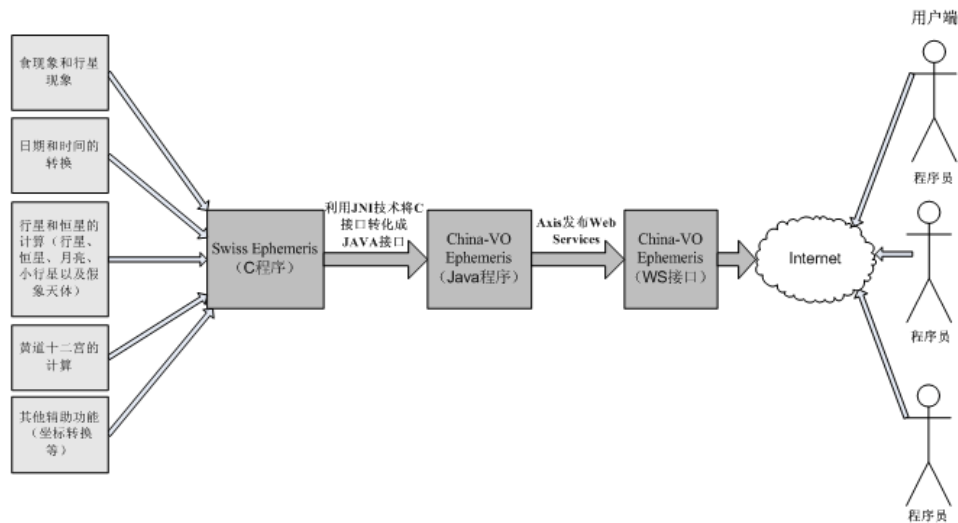


图 5.1: China-VO Ephemeris WS功能结构图

5.1.2 VO-DAS的功能扩展

5.1.2.1 OGSA-DAI中的Activity

OGSA-DAI是通过Activity [35]来功能扩展的。我们要将China-VO Ephemeris WS 计算平台的功能扩展到VO-DAS中, 我们需要编写对应的Activity, 来完成两者的对接。写一个最基本的Activity, 至少需要四步:

1. 定义XML的Schema.

每一个Activity都需要一个XML-Schema, 用来描述Activity的XML的元素结构, 定义Activity的输入输出以及Activity的所有配置细节。Schema中定义的Activity元素被用于执行文档(perform documents)中来指出哪些Activity将被数据资源服务执行。

书写XML-Schema, 可以通过OGSA-DAI提供的XML-Schema模板 [36]来完成。

2. 写一个Java执行类

每一个Activity都需要一个Java执行类, 用来执行和Activity相关联的行为Activity的执行类。OGSA-DAI规定了Activity的执行类的标准生命周期规范(如图5.2)。

书写Java执行类, 可以通过OGSA-DAI提供的Java执行类模板 [37]来完成。

3. 写出对应的客户端工具类

如果客户端想调用服务器端的Activity, 需要写一个和服务器端Activity对应的客户端类, 这个类主要负责生成写入执行文档(perform document)的Activity元素。

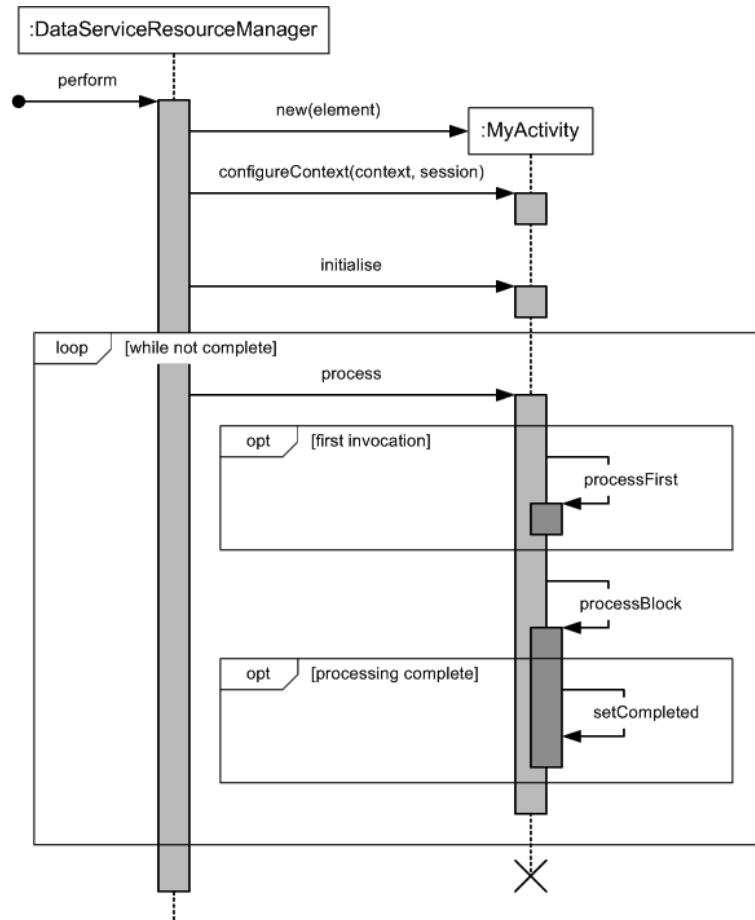


图 5.2: Activity生命周期的序列图

书写客户端Activity工具类,可以通过OGSA-DAI提供的客户端Activity工具类模板 [38]来完成。

4. 安装Activity

一旦完成前三步,我们就可以将创建的Activity的安装OGSA-DAI,详细安装过程可以参见OGSA-DAI的官方网站。

如果将China-VO Ephemeris WS 计算平台成功配置在OGSA-DAI之后,China-VO Ephemeris WS 计算平台就可以作为一个计算节点 (Algorithm Node) 注册,如果注册成功,VO-DAS就可以自动检测到该节点,VO-DAS的客户端发现有相关任务时,就会将任务传达给对应的节点执行。



5.1.3 通过VO-DAS可视化中心食带

5.1.3.1 用户输入方式

用户可以以VO-DAS的统一输入方式：

```
SELECT e.sun FROM swiss:eclipse e WHERE e.time = '2008-03-12' AND  
e.style='Future' AND e.drawn = ture
```

这条输入语句的意思是：在swiss计算节点上，查找2008年3月12日之后最近一次发生日食的情况，并将其可视化。

5.1.3.2 预期结果

输入语句：

```
SELECT e.sun FROM swiss:eclipse e WHERE e.time = '2008-03-12' AND  
e.style='Future' AND e.drawn = ture
```

预期结果（如图5.3）：

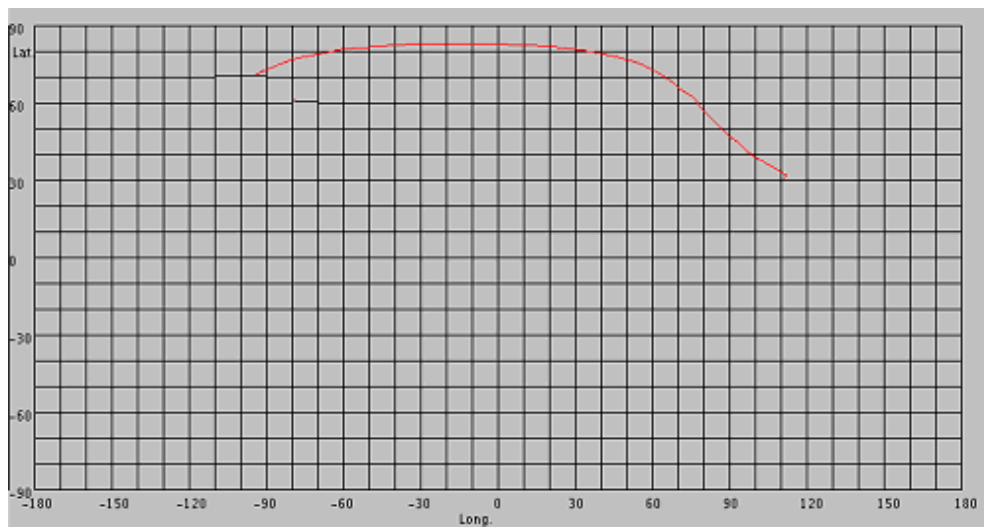


图 5.3: 2008年03月12日之后，第一次中心食食带

输入语句：

```
SELECT e.sun FROM swiss:eclipse e WHERE e.time = '2006-08-12' AND  
e.style='Past' AND e.drawn = ture
```

预期结果（如图5.4）：

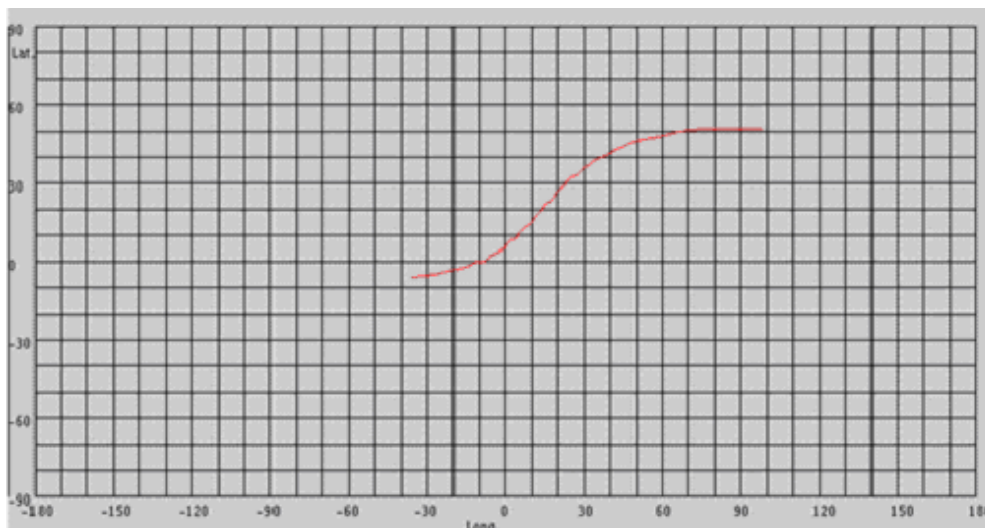


图 5.4: 2006年08月12日之前, 最后一次中心食食带

5.2 VO-DAS其他应用

VO-DAS的应用将侧重于数据查询、数据分析以及可视化服务等, 目前VO-DAS在海量天文数据处理中已经显示了自己的特色:

在采用Sloan数字巡天星表 [28]中的恒星星表作为数据源, 通过研究这些恒星在空间分布密度的特点寻找银河系的卫星星系或球状星团 [29]的科学范例中, 客户端程序会向VO-DAS发出数百个空间位置查询, 每得到一个空间位置的数以千计的查询结果就对这些恒星的空間密度、亮度、颜色特征等进行分析, 给出可视化结果供人工判别; 在对Sloan数字巡天星表的星系红移做出估计 [30], 并使用估计的结果完成星系大尺度结构的可视化的科学范例中, 客户端需要向VO-DAS以及DataNode请求超过2400万条星系位置和红移数据, 并利用这些数据完成星系大尺度结构的绘图。

这些科学范例证实了VO-DAS的科学性、实用性和便利性。随着VO-DAS的不断完善, VO-DAS将在今后天文学家的科学研究和数据处理过程中发挥越来越大的作用。

第六章 总结与展望

虚拟天文台数据访问系统 (VO-DAS) 的提出不到一年的时间,但它是中国虚拟天文台研发团队几年探索和实践的结晶,作为VO-DAS核心模块——任务调度设计与实现,本论文主要包括以下内容:

- ◆第一章主要介绍了VO的产生背景、发展现状以及VO-DAS的产生及其意义;
- ◆第二章主要综述了与VO-DAS相关的关键技术和概念以及IVOA的相关规范文档;
- ◆第三章分别从框架和功能模块上阐述了基于WSRF的VO-DAS;
- ◆第四章主要阐述了VO-DAS的任务调度:分别对VO-DAS的调度方案、Session、生命周期、资源销毁、状态监控、异常处理等具体问题进行了详细地探讨,并从多个角度给出了系统的设计图;
- ◆第五章以VO-DAS对China-VO Ephemeris WS 计算平台的扩展为例,介绍了VO-DAS外布接口的扩展方法,以及VO-DAS在星历计算方面的应用,并简要阐述了VO-DAS在其他方面的科学应用。

目前,VO-DAS仅实现了它的最基本的功能,即支持同步、异步方式的单星表数据查询或多星表数据的联合查询,以及包括数据传输、存储、转换等在内的辅助功能。在以Sloan数字巡天星表作为数据源寻找银河系中的球状星团、估计星系红移等科学范例中,我们证实了以OGSA-DAI为核心的VO-DAS系统对于支持天文学的数据访问是可行性,并证实运行性能在有适当的软硬件环境保障的情况在下可以满足大多数天文学研究的要求。作为比较,我们可以同时支持数百万条数据的异步、同步方式访问;而NVO的OpenSkyQuery实验原型仅能够支持最多5000条数据的同步访问,且不支持数据的异步访问。

但是,VO-DAS目前只是迈出了关键的第一步,离其最终目标尚相差甚远:图像数据和光谱数据的访问、数据挖掘、可视化、天文科普教育等功能模块;系统的安全、数据存储节点(VOSpace)、工作流(WorkFlow)等组件模块尚待完成。

经过China-VO工作团队的一致努力,VO-DAS定将茁壮成长,它最终将成为中国天文数据尤其是即将投入使用的LAMOST巡天数据对外的一个窗口(VO-enabled LAMOST)、天文学家无缝透明地访问全世界的天文数据互操作平台、良好的公众天文教育平台。

附录 A JDL的xml表述形式

JDL/s形式等价转换成JDL/x形式:

```
<?xml version= "1.0" encoding= "UTF-8" ?>
<PROJECT ID= "cc" name= "cc" >
<DESCRIPTION/>
<JOB ID= "gettable" name= "gettable" type= "job" >
<DESCRIPTION/>
<FUNCTION ID= "main" name= "main" >
<DESCRIPTION/>
<STATEMENT>
<OPERATOR ID= "assign" type= "assign" >
<VARIABLEREF ref= "t" />
<FUNCTIONCALL ref= "query" >
<PARAMETERS>
<PRIMITIVEB value= "select glon,glat, j_m, h_m, k_m from TwoMass
where glon>=270 and glon<271 and glat>-10 and glat<10" />
</PARAMETERS>
</FUNCTIONCALL>
</OPERATOR>
<OPERATOR ID= "assign" type= "assign" >
<VARIABLEREF ref= "t" />
<FUNCTIONCALL ref= "addcol" >
<PARAMETERS>
<VARIABLEREF ref= "t" />
<PRIMITIVEA value= "5" />
<PRIMITIVEB value= "h-k" />
<OPERATOR ID= "subtraction" type= "subtraction" >
<VARIABLEREF ref= "t" >
<PRIMITIVEB value= "h_m" />
</VARIABLEREF>
<VARIABLEREF ref= "t" >
<PRIMITIVEB value= "k_m" />
```



```
</VARIABLEREF>
</OPERATOR>
</PARAMETERS>
</FUNCTIONCALL>
</OPERATOR>
<OPERATOR ID= "assign" type= "assign" >
<VARIABLEREF ref= "t" />
<FUNCTIONCALL ref= "addcol" >
<PARAMETERS>
<VARIABLEREF ref= "t" />
<PRIMITIVEA value= "6" />
<PRIMITIVEB value= "j-h" />
<OPERATOR ID= "subtraction" type= "subtraction" >
<VARIABLEREF ref= "t" >
<PRIMITIVEB value= "j_m" />
</VARIABLEREF>
<VARIABLEREF ref= "t" >
<PRIMITIVEB value= "h_m" />
</VARIABLEREF>
</OPERATOR>
</PARAMETERS>
</FUNCTIONCALL>
</OPERATOR>
</STATEMENT>
</FUNCTION>
</JOB>
<JOB ID= "cchist" name= "cchist" type= "job" >
<DESCRIPTION/>
... ..
</JOB>
</PROJECT>
```

参考文献

- [1] 崔辰州. 《中国虚拟天文台系统设计》. 博士论文, 中国科学院研究生院, 2003.
- [2] LAMOST Large Sky Area Multi-Object Fiber Spectroscopic Telescope. <http://www.lamost.org>.
- [3] 刘波. 《中国虚拟天文台的数据访问与互操作研究》. 硕士论文, 中国科学院研究生院, 2006.
- [4] [IVOA] <http://www.ivoa.net/>.
- [5] [AstroGrid] <http://www.astrogrid.org/>.
- [6] [SkyQuery] <http://www.skyquery.net/>.
- [7] [OpenSkyQuery] <http://www.openskyquery.net/>.
- [8] [OGSA] Open Grid Services Architecture. <http://www.gridforum.org/documents/GWD-I-E/GFD-I.030.pdf/>.
- [9] [GGF] Globe Grid Forum. <http://www.ggf.org/>.
- [10] [WS] W3C Web Services Activity. <http://www.w3.org/2002/ws/>.
- [11] [WSDL] Web Service Description Language. <http://www.w3.org/TR/wsdl/>.
- [12] [SOAP] Simple Object Access Protocol. <http://www.w3.org/TR/soap/>.
- [13] [OGSI] Open Grid Services Infrastructure. http://www.globus.org/toolkit/draft-ggf-ogsi-gridservice-33_2003-06-27.pdf/.
- [14] [WSRF] OASIS Web Services Resource Framework. <http://www.oasis-open.org/committees/wsrp/>.
- [15] [GT] Globus Toolkit. <http://www.globus.org/>.
- [16] [OGSA-DAI] Open Grid Service Architecture Data Access Integration. <http://www.ogsadai.org/>.



- [17] [ADQL] Astronomy Data Query Language.
<http://www.ivoa.net/internal/IVOA/IvoaVOQL/ADQL-0.91.pdf>.
- [18] [SIA] Simple Image Access. <http://ivoa.net/Documents/latest/SIA.html>.
- [19] [SSA] Simple Spectral Access. <http://ivoa.net/Documents/latest/SADimEq.html>.
- [20] [FITS] Flexible Image Transport System. <http://fits.gsfc.nasa.gov>.
- [21] [VOTable] <http://ivoa.net/Documents/latest/VOT.html/>.
- [22] 赵永恒. 互联网时代的天文学革命——虚拟天文台. Science. 2002.
- [23] [LSST] Large Synoptic Survey Telescope. 大口径综合巡天望远镜.
<http://www.lsst.org/>
- [24] Chao Liu, Haijun Tian, Dan Gao. *China-VO DAS: Integrated Access of Distributed and Heterogeneous Astronomical Data Resources*. Chinese Journal of Astronomy & Astrophysics. submitted, 2007.
- [25] [SDSS] Sloan Digital Sky Survey. *Sloan 数字巡天*. <http://www.sdss.org>
- [26] [TWO MASS] Two Micron All Sky Survey. *2微米全天巡天*.
<http://www.ipac.caltech.edu/2mass>
- [27] Liu, C. et al. *An astronomical data mining application framework for virtual observatory*. Proceedings of the SPIE, 2006, 6274:627415.
- [28] York, D. G., et al. *The Sloan Digital Sky Survey*. Technical Summary, *Astronomical Journal*, 2000, 120:1579-1587.
- [29] Liu, C., Hu, J. and Zhao, Y.. *Ten Overdensities and an Arc Structure in the Galactic Halo*. arXiv:astro-ph/0612173.
- [30] Wang, D., Zhang, Y., Liu, C. and Zhao Y. *Kernel Regression For Determining Photometric Redshifts From Sloan Broadband Photometry*, *Monthly Notices of the Royal Astronomical Society*. Monthly Notices of the Royal Astronomical Society. submitted. 2007.
- [31] 田海俊、赵永恒等. 基于Web Services的星历计算服务平台设计及其实现. 《国家天文台台刊-天文技术与研究》, 2007年第三期.



- [32] 刘高潮. 虚拟天文台天文教育平台研究. 硕士论文, 华中师范大学, 2004.
- [33] [JNI] Java Native Interface. <http://java.sun.com/j2se/1.4.2/docs/guide/jni/index.html>.
- [34] [Programming interface to the Swiss Ephemeris] <http://www.astro.com/swissep/ephprg.htm>.
- [35] [OGSA-DAI Activity] <http://www.ogsadai.org/documentation/ogsadai-wsrf-2.2/doc/tutorials/activitywrite/index.html>.
- [36] [Activity XML Schema Template] <http://www.ogsadai.org/documentation/ogsadai-wsrf-2.2/doc/tutorials/activitywrite/ActivitySchemaTemplate.html>.
- [37] [Activity Java Template] <http://www.ogsadai.org/documentation/ogsadai-wsrf-2.2/doc/tutorials/activitywrite/ActivityJavaTemplate.html>.
- [38] [Client Toolkit Activity Template] <http://www.ogsadai.org/documentation/ogsadai-wsrf-2.2/doc/tutorials/activitywrite/ClientToolkitActivityTemplate.html>.

发表文章目录

- [1] 田海俊、赵永恒、崔辰州、郑小平. 基于Web Services的星历计算服务平台设计及其实现. 《国家天文台台刊-天文技术与研究》, 2007年第三期.
- [2] Chao Liu, Haijun Tian, Dan Gao. China-VO DAS: Integrated Access of Distributed and Heterogeneous Astronomical Data Resources. Chinese Journal of Astronomy & Astrophysics. submitted, 2007.
- [3] 周霞、田海俊、郑小平等. 磁场中夸克物质的黏滞系数. 高能物理与核物理, 2007 Vol.31 No.1 P.38-42.

致 谢

许久之前就在想，应该在毕业论文中写下些许坚实的文字和激昂的篇章，不过放下笔来沉思，发现能够写下的仍然只是一些平淡的叙述。转眼间已经度过人生似金的研究生三年时光，可是，终究能给养育、培育和关心自己的人们以什么样的交代呢？就是这篇论文吗？似乎不能这样说。然而，于人也罢，于己也罢，必要的感谢总是需要的。

研究生三年能够得到三位名师的指导，实乃人生之幸事，三位导师分别为华中师大郑小平教授、国家天文台赵永恒研究员以及崔辰州副研究员。

首先要真诚感谢我的导师郑小平教授。三年来，郑老师一直都关心和帮助我的学习和生活。学术上，他用勤劳、严谨、认真的作风为我树立了榜样；生活上，他的一言一行更体现高尚的人格魅力，让我受益良多。郑老师常从很小的做人做事细节上指导我、教育我，而且还给我提供很多实践机会，教我如何将理论与实践结合。总之，郑老师渊博的学识、严谨的治学态度、一丝不苟的工作作风、勇于创新的进取精神、灵活的处事原则等等，这一切都让我受益匪浅，在此表示最衷心的感谢。

感谢我在国家天文台两年的导师赵永恒研究员。赵老师知识渊博，待人热诚、宽厚、平易近人。他博采众长，善于纳新，能很好把握天文学的发展动向，他思维敏锐、卓识远见，百忙之中的点拨，总能使我豁然开朗。赵老师，谢谢您！

感谢国家天文台崔辰州副研究员。崔辰州老师是中国虚拟天文台项目的负责人，同时也是我在国家天文台的指导老师。他积极了解虚拟天文台最新动态，每周组织我们进行小组讨论，定期检测我们的工作情况，并及时给予指导。他敏捷的思维，突出的组织能力、认真的工作态度、广泛执着的兴趣爱好，潜移默化着我。在此表示我深深的谢意！

感谢国家天文台的师兄刘超博士，在国家天文台两年的学习和生活中，超兄给予我师长般的指导和照顾，在论文撰写的过程中，他给了我很多的帮助与建议，使我可以更加顺利地完成学业。

感谢China-VO项目组、LAMOST项目组的所有兄弟姐妹：尹红星、王丹、罗宇、杨阳、刘健、郑征、孙华平、路勇、高丹、李丽丽、吴悦、孙士卫、李乡儒、刘中田等，以及海外求学的吴潮博士、张旭同学，和你们相处的日子里，我十分开心。

感谢国家天文台的张彦霞、罗阿里、张昊彤、石火明、陈建军等老师，感谢他们的关心和帮助，同时感谢国家天文台邓李才研究员对我论文的指正。



感谢华中师大的同门兄弟姐妹：杨书华、刘学文、康缈、潘娜娜、周霞、刘高潮、吴娟、彭俊金、俞云伟、皮春梅、陈俊、张黎、彭玲等等，有你们的陪伴，自己仿佛置身于一个和睦的大家庭中，总有欢声笑语相伴。

感谢华中师大物理学院王恩科院长、刘连寿老师、杨丕博老师、陈义成老师、杨纯斌老师、吴少平老师、乔翠兰老师、高闯老师等，你们的课讲得很精彩！

感谢华中师大和我朝夕相处的同窗好友：刘磊、丁亨通、罗俊杰、刘强、刘金平、雷兵、宋艳生等，以及已经踏上工作岗位的韦进、张烈、范夫伟等，和你们相处，我永远开心。

感谢华中师大教务处吴绍靖老师、曹惠东副处长、国际文化交流学院任友洲副院长等，在华师学习期间，给你们添了不少麻烦！

特别感谢我勤劳而质朴的双亲，他们为我的成长和求学操劳二十余载，他们的汗水和艰辛铺满了我路过的每一个脚印；感谢我的两个姐姐田海英、田海辉以及姐夫胡才东、王道峰在学业上的鼎力支持和生活上无微不至的关怀照顾；感谢我的女友吴小艳女士，感谢一直以来她对关心以及在我的学业的理解。

感谢所有关心、照顾、帮助过我的亲人、朋友、老师、同学……

今后更多的成就将是你们最好的回报。

于国家天文台实验楼324室

二零零七年五月十六日