第 34 卷 第 5 期 计 算 机 工 程 2008 年 3 月
Vol.34 No.5 Computer Engineering March 2008
·软件技术与数据库· 文章编号：1000—3428(2008)05—0045—03 文献标识码：A 中图分类号：TP391

改进空间信息服务质量的设计策略

邹志强[1,2,3]，江 南[2,3]，张 正[3,4]
(1. 南京邮电大学计算机学院，南京 210003；2. 中国科学院南京地理与湖泊研究所 GIS 研究室，南京 210008；
3. 中国科学院研究生院，北京 100080；4. 中国科学院软件研究所软件工程中心，北京 100080)

摘 要：针对分布式空间信息领域服务的特点，结合“国家科学数据共享工程”的开发实践，提出了改进空间服务质量的 3 种策略：针对组件纵向关系的设计，采用面向服务结构和中心元数据服务器对 OGC 服务框架的改进策略；针对组件横向关系的设计，引入面向方面编程的策略；针对空间信息的领域特点，采用地图服务和空间数据缓冲池的设计策略。给出了一个原型系统实现的拓扑结构和相应的测试数据，验证了 3 种策略优越性。

关键词：面向服务的框架；面向方面编程；空间信息 Web 服务

Design Strategies to Improve QoS of Spatial Information Web Services

ZOU Zhi-qiang[1,2,3], JIANG Nan[2,3], ZHANG Zheng[3,4]
(1. College of Computer, Nanjing University of Posts & Telecommunications, Nanjing 210003; 2. GIS Lab, Nanjing Institute of Geography & Limnology, Chinese Academy of Sciences, Nanjing 210008; 3. Graduate School, Chinese Academy of Sciences, Beijing 100080; 4. Software Engineering Center, Software Institute, China Academy of Sciences, Beijing 100080)

【Abstract】According to the characteristic of distributed Spatial Information Web Services(SIWS), this article outlines three strategies for improving quality of spatial services based on the developed practice of “national science data sharing project”. The strategies are as following: improving the framework of OGC (Open Geospatial Consortium) by introducing service-oriented architecture and metadata server; the introduction of aspect oriented programming into SIWS; the strategy for implementing Web Map Service (WMS) and spatial data buffer in spatial information domain. It presents the topological structure of our prototype and some corresponding test data to verify these three strategies.

【Key words】service-oriented architecture; aspect oriented programming; spatial information Web Services

空间信息Web服务是指通过计算机网络查询、获取、交换和再加工与空间实体相关的数字化信息服务，包括对部分信息处理的透明使用和互操作。目前的空间数据共享和信息服务平台，一些是采用数据交互中心的形式，一般是基于元数据的目录服务，可为用户提供按关键字查询的空间信息等服务，但是这种服务的实现往往是紧耦合的，在多个系统之间缺乏互操作性。Web服务技术为解决这个问题提供了可能[1]，本文采用它来提供基本的空间信息Web服务。这种空间信息Web服务(Spatial Information Web Services, SIWS)是运行在Web上的自包含、模块化的应用程序，可以在网络中被描述、发布、查找以及调用，实现了软件的动态提供。

1 对 OGC 服务框架的改进

面向服务的架构(Service-Oriented Architecture, SOA)可改善分布式环境中系统的耦合性。而OGC(Open Geospatial Consortium)给出了一个权威的参考规范[2]，本文在构建SIWS框架时与SOA相结合，并引入中心元数据服务器，对OGC的服务框架进行改进，改进后的SIWS框架如图 1。

该框架包含几个主要组成部分：Data Source Input, Feature Services, Imagery Services, Services Provider 以及用于元数据服务的 Metadata Server。其中 Data Source Input 又可以分为 Legacy App(遗留系统)和 Interop App(互操作的应用)。这两部分是按照分层结构设计的，负责处理各种各样的输入数据，例如属性数据、栅格的遥感数据以及矢量的空间数据。

在互操作的应用部分又添加了数据服务器和关系对象的映射层，本系统采用 Hibernate 来实现这种 O/R 映射。同时考虑到和 OGC 的兼容，设计了统一的服务接口，这些接口都跟中心的 Metadata Server 和其他平行的系统相连。框架中 Feature Services 负责处理跟 WFS(Web Feature Service)要素服务相关的空间信息服务；Imagery Services 负责处理和 WMS(Web Map Service)Web 地图服务相关的过程；Services Provider 负责向外提供各种经过空间信息平台处理后的服务；Metadata Server 负责提供整个系统的统一数据规范。从图 1 中可看出这些组件的设计策略和 Data Source Input 的类似，都是按纵向分层实现的，只是具体业务不同，详述见文献[3]。

基金项目：国家科技部项目(2004DKA20180)；南京邮电大学科研项目(NY206039)
作者简介：邹志强(1967－)，男，讲师、博士研究生，主研方向：软件工程，分布式 GIS；江 南，研究员、博士生导师；张 正，博士研究生
收稿日期：2007-03-10 **E-mail：**zzq2006@njupt.edu.cn

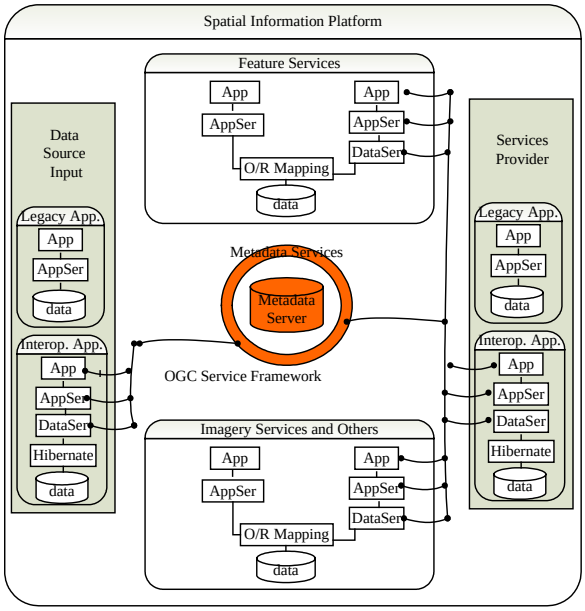

图 1　空间信息服务框架

2　AOP 策略的引入

面向方面编程 (Aspect Oriented Programming, AOP)提供了一种解决软件开发中分离关注点的途径[3]。GoF(Gang of Four)设计模式追求的是调用者和被调用者之间的解耦，AOP也是这种目标的一种实现，它更多地是强调对象之间的继承性。但是，在分析图 1 中一个典型的服务(如 Imagery Services)实现框架之后，不难发现它的一个不足，即在建立对象按上下方向分层时比较清晰，例如可以分为应用层、应用服务器层、数据持久层和数据层，可是在对应用服务器层的具体实现时显得不足，它没有把实现的核心业务逻辑对象(如影像数据的查找、图像的合并等)和其他不具有继承关系的业务对象(如访问控制权限、日志的记录和同步的处理等)分开，下面引入AOP进行优化。

2.1　未采用 AOP 策略的设计

在系统开发初期，规模不是很大时，本文参考了图 1 分层来划分模块并编码实现，例如，在实现 Imagery Services 时，伪代码 1 如下：

```
package org.deegree_impl.services.wms.capabilities;
public class DataSource_Impl implements DataSource, Marshallable{
1.doSecurityCheck();//安全验证
2.logging("DataSource start");//执行前日志记录
3.String getName(){…}//returns datasource name
4.OGCWebService getOGCWebService() {…}
//returns an instance of the OGCWebService
5.WMSGetMapRequest getGetMapRequest() {…}
//returns an instance of a WMSGetMapRequest
6.logging("DataSource end");//执行后日志记录
}
```

为减少图 1 其他服务模块中不具有继承关系的业务对象重复，如上述伪代码中的第 1 行、第 2 行、第 6 行，把这些模块之间公共的“方面”(aspect)封装起来，减少系统的重复代码，同时降低模块间的耦合度。

2.2　采用 AOP 策略的设计

为描述的一致性，先定义下面 3 个术语：

定义 1 (关注点) 一个关注点可以是特定的问题、概念或是应用程序的兴趣点，例如伪代码 1 中的第 1, 2, 6 行的安全验证、日志记录等都是关注点。

定义 2 (方面) 一个方面是对一个横切关注点的模块化，它将那些原本散落在各处的、用于实现该关注点的代码规整在一处，它表示的是多个对象间横向的关系。

定义 3(建议) 一个建议是“方面”执行的具体逻辑，可用拦截器(interceptor)形式来表现。

本文采用 AspectJ 框架来实现 AOP，这是一种开源的轻量级框架，它使用 Java/AspectJ 代码。

采用 AOP 对伪代码 1 优化后的伪代码 2 如下：

```
package org.springframework.aop.wms.capabilities;
import org.springframework.aop.support.*;
public class DataSource_Impl
extends DelegatingIntroductionInterceptor
implements DataSource, Marshallable,IValidatable{
1.BusiLogic business =new BusiLogicCoreConcern();
2.BusinessLogic = busiObj
    (BusinessLogic)LoggingProxyAspect.bind(
     SecurityProxyAspect.bind(business));
3. busiObj.getName(){…}//returns datasource name
4. busiObj.getOGCWebService() {…}
//returns an instance of the OGCWebService
5. busiObj.getGetMapRequest() {…}
//returns an instance of a WMSGetMapRequest
}
```

使用了 AspectJ 以后的核心业务代码简洁，伪代码 2 中的第 1 行、第 2 行为使用动态代理模拟安全和日志的横切 aspect，伪代码 1 中的 aspect 似乎看不到了，因为被分离到了伪代码 3 中形成了公共模块，伪代码 3 如下：

```
public class SecurityProxyAspect implements InvocationHandler {
1.static Object bind(Object obj) {…}//生成对象代理
2.Object invoke(Object proxy, Method method, Object[] args) {…}//截获所有对象的方法调用
3.void doSecurityCheck() {}//横切关注点：安全验证
4.void doLogging(){}//横切关注点：日志记录
}
```

综合 3 段伪代码可以看出，调用模块和被调用模块之间得到了解耦，参考文献[4]的计算，系统的代码量、开发时间和调试时间这 3 个方面分别可以减少约 14%, 33%和 70%，系统性能得到了提高。

3　空间信息领域服务实现的策略

前文所述的策略具有一定的通用性，本节则是针对SIWS领域讨论优化的策略。在SIWS中，WMS, WFS和WCS(Web Coverage Service)是 3 个主要的空间信息服务，而WMS又常以WFS和WCS为数据源最终生成地图影像提供给用户[5]。本文假设一次Web服务表示为WSi(Ii,Oi)，其中，WSi是Web服务的名字；Ii 是请求该Web服务的输入；Oi是该Web服务输出的对象。其计算模型WMS，对应数据模型用XML表达，如：wms_capabilities.xml。核心算法如下：

输入 Ii　一个 Web 服务的请求

输出 Oi　服务输出的结果对象，它可以是一个字符文件，也可以是一组二进制流表示的图像文件

步骤 1 WMS Servlet 的初始化。

步骤 1.1 如果没有该 WMS Servlet 初始化实例，则先初始化，然后转向步骤 1.3。

步骤 1.2 如果有该 WMS Servlet 的初始化实例，则从 WMServicePool 中得到一个 Servlet 实例。

步骤 1.3 调用 doGet(HttpServletRequest req, HttpServlet Response resp)方法，对请求进行处理。

步骤 2 调用缓冲池管理模块 BufferManager()。

步骤 3 如果缓存中存在该数据模型对应的影像数据，则无需再调用计算，从缓存中取该影像数据，转步骤 5.1。

步骤 4 如果缓存中没有对应的影像数据，则调用计算模型求解具体的数据。

步骤 4.1 根据金字塔算法，调用 TileImageTree()，按照四叉树的数据结构对大数据量遥感数据进行分割。

步骤 4.2 根据 Ii，调用计算模型 performGetMap()，遍历已有的 Raster 切片数据。

步骤 4.2.1 如果有相应的数据，则转步骤 5.1。

步骤 4.2.2 如果没有相应的数据，则转至步骤 5.2。

步骤 5 输出结果，返回。

步骤 5.1 如果有满意结果，则成功返回一张影像图。

步骤 5.2 如果没有满意结果，则抛出错误信息。

具体查找数据的过程见图 2。

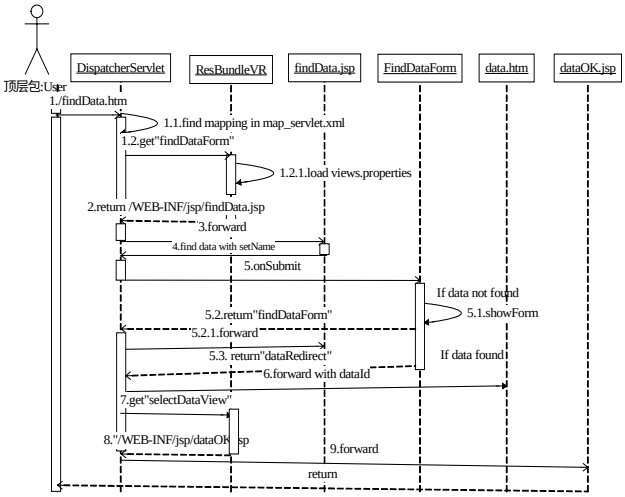

图 2 在数据服务时 UML 序列图

算法说明：

(1)步骤 1 中缓冲池 WMServicePool 的作用是优化 WMS 的服务质量。

(2)步骤 2 中缓冲池管理模块的设计是为提高服务质量，因为在空间信息领域里，经常涉及到大数据量的处理。SIWS 策略：根据用户请求的空间数据本身具有的连续性，系统常常要处理用户请求的同分辨率/经度纬度相近数据、相同经纬度但分辨率更高的数据，以及最近访问过的数据，缓冲池管理模块将对这 3 类数据预先处理并存储在空间数据缓冲池中。这样用户下一次请求很大几率上就不需要再访问后台的 Web 服务了，降低了系统相应时间，提高了 Web 服务的质量。数据缓存初始化为 100 条，30 分钟动态刷新，可设置 MaxBuffer 的大小。

(3)方法 TileImageTree()可管理大数据量的遥感图像，避免了对大数据的直接访问，节省了系统的内存。它可以根据调用参数，生成不同数量和分辨率的多个切片，例如在第 4 节对 WMS 的测试中包含 0.001, 0.000 5 这两个分辨率的共 20 张切片。

(4)方法 performGetMap()是根据 WSi(Ii,Oi)中的 Ii，从(3)生成的多个切片中查找匹配的切片数据。

4 原型的测试

4.1 测试的系统结构和环境

为了验证 SIWS 的可行性与有效性，笔者开发了一个原型系统，具体的网络拓扑结构如图 3。

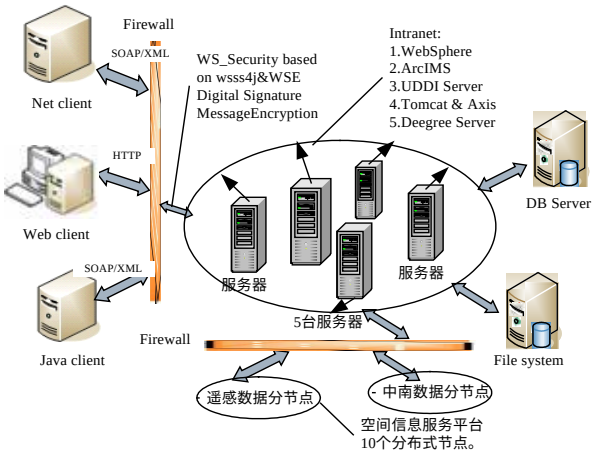

图 3 空间信息服务平台的网络拓扑结构

整个实验是在如下硬件环境中进行的：1 台联想万全 R520-5122 服务器和 PC 测试机器。整个实验主要用到如下软件环境：Windows 2003, IBM WebSphere v5.1, ESRI ArcIMS 9.0, Oracle9i, ESRI ArcSDE,Tomcat5.x,AXIS 和 LoadRunner 7.8 等软件。

4.2 测试结果

由图 3 的客户端可知可以满足 Internet 上多种用户的请求。为了能对多个并发用户请求时系统的性能进行测试，采用 LoadRunner7.8，并设置了 10IP“欺骗”策略，以防止系统对同一个 IP 有缓存的作用，这样增加了测试压力和模拟测试的真实性。测试结果如表 1。当并发用户数为 39 个，整个测试时间为 136 s，模拟处理 WMS120 次时，此时 Error 这行为 0，当继续增加压力到并发用户数为 40 个时，Error 这行错误数变为 1，系统抛出错误，所以在本文的测试环境下，最大并发用户数为 39 个，参考文献[6]中的实验：最大并发用户数为 15 个，本文测试结果是理想的。

表 1 WMS39-Result

测试的状态	最大值	最小值	均值	方差	最后测试结果
Error	0.00	0.00	0.00	0.00	0.00
Finished	40.00	0.00	20.22	9.67	40.00
Ready	2.00	0.00	0.00	0.00	0.00
Running	39.00	0.00	19.78	9.67	0.00

5 结束语

本文对 SIWS 框架的实现进行了的研究，提出了改善服务性能的 3 种策略，并用把这种策略成功应用到项目设计中，改善了系统的性能。但以下两个方面还需要进一步研究：(1)使用 AspectJ 框架支持了 AOP 的实现，但它缺乏对企业级服务的支持，为此，笔者将会把 Spring 框架引入到 SIWS 中，从而更好地解决企业级服务中的事务性和安全性问题；(2)实现数据服务时，空间信息本体库还须完善，为更智能地检索和共享信息打好基础。

参考文献

[1] 郑 锋, 涂 平, 王钦敏. 基于 WebServices 的政务信息共享平台[J]. 计算机工程, 2006, 32(8): 134-136.

[2] OGC, OpenGIS® Web Services Architecture[Z]. (2006-11-16). http://portal.opengeospatial.org/files/?artifact_id=1320.

[3] Zou Zhiqiang, Jiang Nan, Hu Bin, et al. Research and Implementation of Geosciences Data Sharing Infrastructure Based on SOA & Web Services[C]//Proc. of IEEE International Geoscience and Remote Sensing Symposium. Denver, USA: [s. n.], 2006.

（下转第 50 页）