

doi: 10.3969/j.issn.1006-1576.2011.10.016

一种网络空间数据集成访问方法

石海潜, 陈萃, 刘义

(国防科技大学电子科学与工程学院, 长沙 410073)

摘要: 为了有效地组织管理获取的空间数据, 对网络空间数据集成访问方法进行研究。分析在网络上分布的空间数据集成访问技术, 基于面向服务的体系结构(service-oriented architecture, SOA)建立网络上分布的空间数据集成访问框架, 提出一种客户端对多尺度空间数据的组织调度方法, 并设计一种网络空间数据的访问协议, 基于MapReduce 计算模型设计实现了面向海量遥感影像索引并行构建方法, 并构建一个原型系统进行了验证。实验结果证明: 该方法实现了海量空间数据快速预处理和网络环境下快速集成访问, 验证了集成访问的可行性。

关键词: 面向服务的体系结构; 多尺度; MapReduce; 集成访问

中图分类号: TP311.12 **文献标志码:** A

An Integrating Access Method of Network Spatial Data

Shi Haiqian, Chen Luo, Liu Yi

(College of Electronic Science & Engineering, National University of Defense Technology, Changsha 410073, China)

Abstract: In order to effectively organize and manage the obtained spatial data, this paper studies the integrating access method of network spatial data. Analysis of the integrating access technology of spatial data distributed on network, builds the network spatial data integrating access framework based on service-oriented architecture (SOA) architecture, puts forward a method of organizing and scheduling the multi-scale spatial data, designs a network spatial data access protocol, designs and realizes a method based on MapReduce calculation model which parallel build the mass remote sensing image index, finally constructs a prototype system to validate the proposed methods. Experiment results show that the method realizes the massive spatial data quickly preprocessing and integrating access under the network environment, verify the feasibility of integrating access.

Keywords: service-oriented architecture; multi-scale; MapReduce; integrating access

0 引言

随着空间数据获取和处理技术的飞速发展, 数据量已达到了 GB、TB 甚至 PB 级别。但由于应用环境、应用部门不同, 空间数据的组织与使用方法各不相同, 导致数据异构、存储分布, 形成数据孤岛, 给数据共享使用和数据挖掘带来了极大的困难。因此, 如何组织管理和集成应用这些网络化分布的空间数据, 使之更好地为应用提供服务, 已成为一个亟待解决的重要课题。

面向服务的体系结构(service-oriented architecture, SOA)是一种组件模型, 用于应用程序的部署, 它将应用程序的不同功能单元(称为服务)通过这些服务之间定义良好的接口和契约联系起来, 使得构建在系统中的服务可以以一种统一和通用的方式进行交互。因此, 笔者基于 SOA 体系架构建立了网络空间数据集成访问框架, 并提出一种客户端对多尺度空间数据的组织调度方法。

1 网络空间数据集成访问框架

SOA 中的每个服务会在网络上广播自己的存

在, 用户可以在任何经过适当授权的机器上执行相应的操作。由于每个服务都是一个自己可以控制的组件, 因此它们可以存在于网络上的任何地方, 整个系统不再需要一台专用的服务器来处理请求^[1]。

SOA 网络模型示意图如图 1。

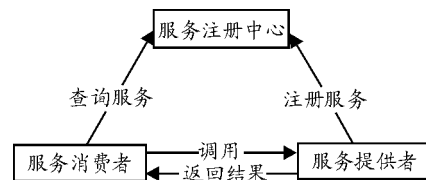


图 1 SOA 网络模型

笔者基于 SOA 体系架构建立了网络空间数据集成访问框架, 如图 2。

该框架下进行网络空间数据访问的具体工作流程为: 在网络环境下, 客户端向服务注册中心查询当前网络上可用的空间数据服务, 获取服务的元数据信息; 经过元数据绑定之后, 客户端可直接定位到空间数据服务器, 在进行数据查询浏览时, 向服务器提交数据请求索引, 服务器接收请求后, 解析用户的数据请求, 获取客户端用户所要检索的数据

收稿日期: 2011-06-24; 修回日期: 2011-07-28

基金项目: 国家 863 项目基金“空天信息海量数据管理与动态更新技术”(2008AA12A211); 国家 863 项目基金“地理计算引擎与开放式地理计算构建环境”(2011AA120306)

作者简介: 石海潜(1984—), 男, 广西人, 壮族, 硕士, 从事信息处理与信息系统技术研究。

参数，通过一定的检索技术在服务器中快速定位所要访问的数据集，在将数据调度到服务器缓存中的同时，对数据进行必要的预处理(如矢量栅格化、格式转换、数据压缩等)，而后直接传送回客户端；客户端将服务器上的数据下载到本地缓存中，并完成对数据的渲染绘制。

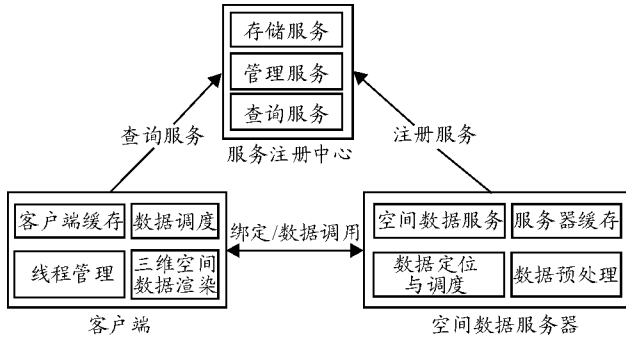


图 2 网络空间数据集成访问框架

2 面向网络的三维空间数据集成访问关键技术

2.1 客户端数据组织方法

1) 基本组织方法

要在客户端实现对分布式的空间数据进行实时流畅的集成显示，数据的组织与调度管理起了决定性的作用。针对当前普通 PC 机内存资源有限，而空间数据量巨大的特点，笔者采用细节层次 (level of detail, LOD) 模型作为多尺度空间数据的可视化方式，在建立地表模型时，充分考虑人的视觉特点，对处于视觉不同区域的模型采用不同的分辨率，在近处观察物体时采用精细模型，在远处则采用较为粗糙的模型，整个地表模型以“块”为单位进行显示，从而具有多分辨率的细节层次^[2]，如图 3。



图 3 LOD 模型

按照上述思路，客户端采用分层、分块的影像金字塔技术对数据进行组织。影像金字塔的思想是把原始高分辨率、大数据量的影像作为金字塔的最底层，通过对原始影像采用特定的重采样方法，建立一系列范围相同但详尽程度和分辨率不同的影像，其中原始影像分辨率最高，经重采样得到的影像随着金字塔层数的增加分辨率逐渐降低。显示时根据分辨率的具体要求调用影像金字塔对应层的图

像块，从而实现快速显示、漫游的目的。图 4 为影像金字塔模型示意图。

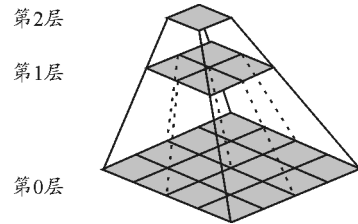


图 4 影像金字塔模型

2) 索引构建方法

首先按照 Plate Carree 投影格网四叉树 (plate carree projection grid quadTree, PCPGQT)^[3] 层次剖分模型建立全球剖分格网，PCPGQT 层次剖分模型是采用 Plate Carree 投影将地理坐标经纬度 $[-180^\circ, -90^\circ]$ 到 $[90^\circ, 180^\circ]$ 范围内的地球表面投影成一个长宽比为 2:1 的规则矩形平面，以此为底面构建离散多分辨率影像金字塔，然后对各层进行均匀划分，将投影得到的矩形平面划分为 N 行 M 列，为了使划分得到的影像块为正方形，通常取 $M=2N$ 。

采用 PCPGQT 剖分后得到的地图平面如图 5。对剖分得到的每一分辨率层次上的影像块采用线性四叉树的思想，使用行号加列号的方式进行编码(如图 6)。具体而言，对于第 0 层，即金字塔顶层的图像按角度等分为 n 行 $2n$ 列 ($n \geq 1$)，以 $(-180^\circ, -90^\circ)$ 为原点，向北和东为正方向，原点所在的行为第 0 行，所在的列为第 0 列。根据多分辨率金字塔的组织方式，第 1 层会被划分为 $2n$ 行 $4n$ 列，第 k 层划分为 $2^{k-1} \cdot n$ 行 $2^k \cdot n$ 列；各层上影像块的编码方式与第一层相同。

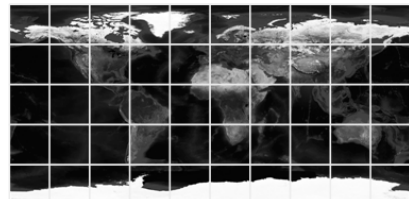


图 5 PCPGQT 剖分示意图

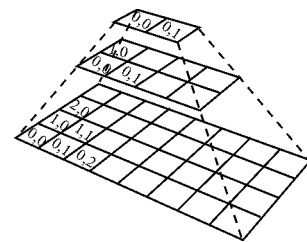


图 6 影像块编码示意图

3) 数据的存储方法

由于客户端需要经常从数据服务器上下载并更

新金字塔的数据瓦片, 因此在客户端采用一种多层次的文件目录结构对金字塔体系进行存储。其具体存储格式如图 7。

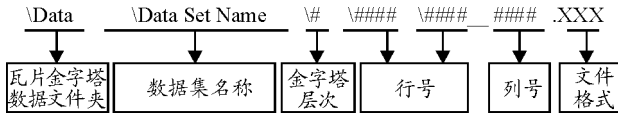


图 7 瓦片金字塔数据存储格式

2.2 数据访问调度

数据访问协议定义了客户端与服务器之间进行数据请求与传送的接口规范。按照现有基于空间信息服务的系统构建方式, 数据访问大多采用符合 OGC 标准的网络地图服务(WMS)实现。但是, 当数据访问并发数较大时, 由于必须实时生成结果数据, 其性能会急剧下降。为此, 笔者基于瓦片金字塔数据结构的想法, 设计了瓦片地图服务协议作为栅格空间数据的访问协议。

瓦片地图服务 TMS 是一种基于元数据的空间数据共享模型, 采用空间数据分布式存储、元数据集中管理的模式, 实现异地空间数据的共享, 其数据共享框架如图 8。

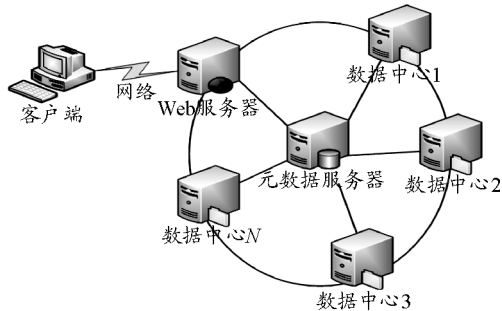


图 8 分布式空间数据共享框架

通过一个集中的元数据服务器, 用户可以快速地从整个元数据库中查询可用的数据集, 数据中心服务器不必为数据检索服务, 仅需根据用户的请求范围, 找到所需的瓦片数据 Tiles 并返回, 这样图片不需要动态产生, 提高了栅格数据访问速度, 从整体上提高了系统的访问效率^[4]。

采用这种方式时, 服务器端需要对遥感影像数据进行预处理, 即按照瓦片金字塔模型进行分层、分块, 构建影像索引。

在本地数据的显示调度环节, 首先根据当前窗口的显示范围, 计算出当前显示的级数, 找到与级数最相近的影像层数据; 然后确定需要调用的具体数据块, 按照预先建好的数据位置索引和邻接关系, 在文件系统中快速找到对应的数据。在影像放大、缩小、漫游的过程中, 根据当前窗口显示范围的变化实时更改调用的数据。为了节省内存空间, 需要

实时监测调入的数据块情况, 将不在显示范围内的数据块从内存中清除, 并调入新的数据块。

2.3 基于 MapReduce 的遥感影像索引并行构建

空间数据包括栅格数据与矢量数据, 对矢量数据进行预处理的方法是使用 GeoServer 或 ArcGIS Server 等进行实时地矢量栅格化, 然后传送给客户端; 而对于栅格数据, 则构建遥感影像金字塔, 最后形成一个多层次的文件夹, 其中存放不同分辨率的数据文件, 对剖分得到的每一分辨率层次上的影像块均采用行号加列号的方式进行编码索引。

在构建遥感影像金字塔时, 传统的算法通常是在单机上运行, 由于受到单机处理能力的限制, 这些算法的效率较低。笔者提出了一种基于 MapReduce 的遥感影像索引并行构建方法, 在集群环境下发挥多台计算机的计算能力, 使数据处理效率得到大幅提高。

MapReduce 是一个针对大规模集群中海量数据处理的并行编程模型, 因此该模型在海量遥感影像数据预处理方面具有天生的优势。用 MapReduce 进行影像金字塔并行构建的具体实现为^[5]:

1) Map 函数: Key 为原始影像的 BoundingBox, Value 为原始影像, 如图 9。

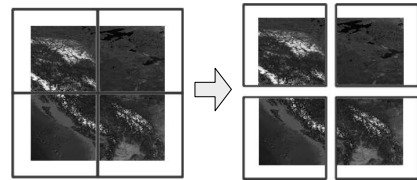


图 9 Map 阶段的数据处理

Mapper 根据 0 层瓦片大小及当前层的层次, 生成相应的标准投影格网, 通过标准投影格网与原始影像的“求交”运算, 将原始影像切割分块并投影到标准投影格网中, OutputKey 为标准投影格网的 BoundingBox, OutputValue 为分块之后的影像。

2) Reduce 函数: Key 为标准投影格网的 BoundingBox, Value 为分块之后的影像, 如图 10。

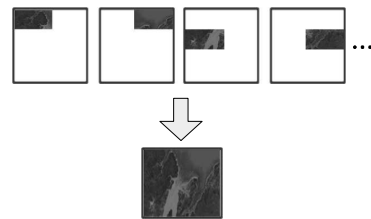


图 10 Reduce 阶段的数据处理

Reducer 根据标准投影格网的 BoundingBox 对分块后影像进行合并。

笔者采用此方法在 IBM 高性能计算平台下(平

台硬件：2 个 IBM X3650M3 管理和 IO 节点，28 个两路四核 IBM BladeCenter HS22 计算节点、4 个 IBM X3550M3 机架服务器，由高速 40 Gb QDR Infiniband 进行网络连接)进行实验，数据使用广州地区 0.3 m 分辨率遥感影像，大小为 8 GB。实验测试了构建的影像金字塔层数对预处理时间的影响，结果如图 11。

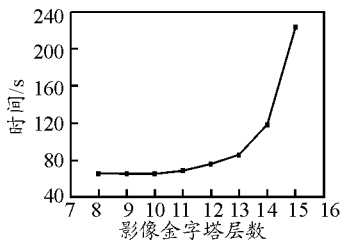


图 11 遥感影像金字塔并行构建性能

据推算，该配置下，笔者提出方法每天可处理的数据量约在 3 TB 左右。

2.4 空间数据服务发布

根据瓦片地图服务 TMS 协议中设计的数据共享框架，空间数据服务的发布主要分为 2 部分：

1) 将预处理得到的数据发布成 HTTP 服务，发布工具可用 IIS、Apache 等。

2) 注册服务，服务注册中心使用 Oracle10g 对元数据进行集中管理，数据库中建立了 2 张表，ServerURL 和 DataSet。其中 ServerURL 表存储数据中心的 URL 地址，每一条记录对应一个数据中心；DataSet 表存储数据集的元数据，每一条记录保存的信息包括数据集的 URL 地址、名称、存储目录、0 层瓦片空间范围大小、瓦片层数、数据格式、地理空间范围等。

3 原型系统

原型系统采用 Windows 操作系统作为开发平台，3D 引擎使用 DirectX，充分考虑系统的可移植性和重用性，基于 .Net Framework 3.5 使用面向对象的 C# 语言开发了系统的三维集成展示框架。

原型系统分为 3 层逻辑结构，分别是数据层、功能层、显示层。具体划分为影像金字塔数据调度模块、三维显示模块、WMS(网络地图服务)导入模块、TMS(瓦片地图服务)导入模块、目标查询定位模块、3D 模型加载模块、动态气象信息加载模块等部分，如图 12。

1) 影像金字塔数据调度模块根据用户对空间数据浏览区域的操作，基于当前窗口范围对数据加载、撤出、替换等进行实时调度。

2) 三维显示模块提供三维数字地球方式对多源空间数据进行浏览。

3) WMS 导入模块遵照 OGC 接口协议，将现有 WMS 服务集成进本系统，使用户在本系统内访问 WMS 提供的网络地图数据服务。

4) TMS 导入模块通过自定义的数据访问协议 TMS，将 TMS 服务器上的数据集导入到本地缓存，支持对数据集的调度。

5) 目标查询定位模块提供文本检索的方式，从目标资源库中查询返回用户感兴趣的目标属性信息。

6) 3D 模型加载模块将文件系统中的 3D 模型加载到系统中进行渲染。

7) 动态气象信息加载模块通过粒子系统技术实现对动态信息的渲染。

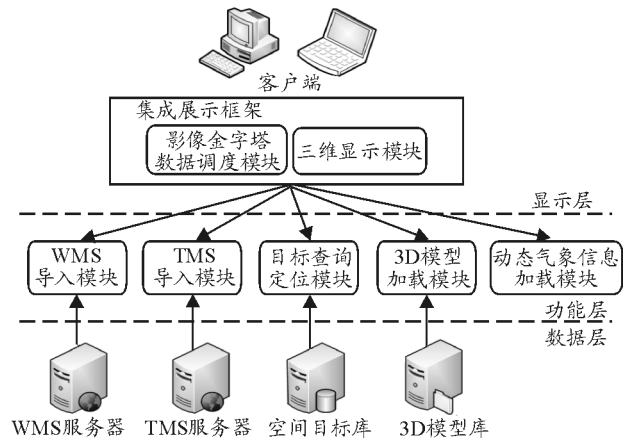


图 12 原型系统逻辑结构

原型系统的界面主要分为 3 个部分，即功能区、图层管理区和三维数字地球展示区，如图 13。

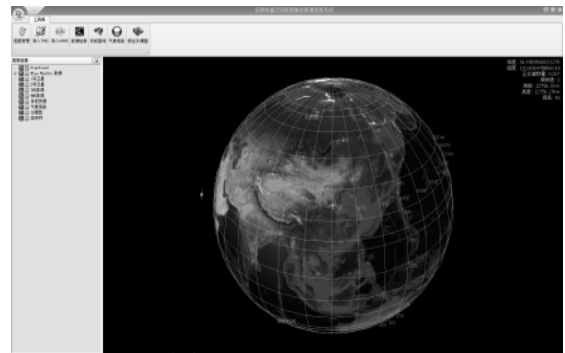


图 13 原型系统界面

其中功能区位于界面上方，工具栏上包含系统各个功能模块对应的动作按钮；图层管理区用于查看当前所有图层，并可对图层进行相关的操作；三维数字地球展示区用于空间数据的三维显示、查询等。图 14~图 19 分别展示了原型系统对网络空间数据进行集成访问的各项子功能。