

# 基于云平台的高并发 WebGIS 服务<sup>①</sup>

王少萍<sup>1,2</sup>, 顾乃杰<sup>1,2</sup>, 沈 婕<sup>3,4</sup>

<sup>1</sup>(中国科学技术大学 计算机科学与技术学院, 合肥 230027)

<sup>2</sup>(中国科学技术大学 先进技术研究院, 合肥 230027)

<sup>3</sup>(南京师范大学 虚拟地理环境教育部重点实验室, 南京 210023)

<sup>4</sup>(南京师范大学 地理科学学院, 南京 210023)

**摘要:** 随着 Internet 技术的发展, 传统的 WebGIS 服务器在面对大用户群高并发访问时会出现服务延迟甚至拒绝服务的现象. 针对该问题, 本文提出了一种基于云平台的应对高并发的 WebGIS 服务器架构. 架构使用云平台为 WebGIS 服务器提供弹性计算和存储资源, 并从负载均衡、缓存设计、数据库集群三方面缓解高并发瓶颈. 选择开源服务器端软件 GeoServer 作为 WebGIS 应用实验部署. 实验数据表明使用缓存机制明显降低了 WebGIS 服务响应时间. 与单物理服务器相比, 云平台 WebGIS 服务器集群架构能够有效处理高并发请求, 且随着集群规模扩展, 云平台 WebGIS 系统能够获得良好的加速比.

**关键词:** 缓存; 云平台; 数据库集群; 高并发; 负载均衡; 网络地理信息系统

## High Concurrent WebGIS Service Based on Cloud Platform

WANG Shao-Ping<sup>1,2</sup>, GU Nai-Jie<sup>1,2</sup>, SHEN Jie<sup>3,4</sup>

<sup>1</sup>(School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China)

<sup>2</sup>(Institute of Advanced Technology, University of Science and Technology of China, Hefei 230027, China)

<sup>3</sup>(Key Laboratory of Virtual Geographic Environment, Nanjing Normal University, Nanjing 210023, China)

<sup>4</sup>(College of Geographic Sciences, Nanjing Normal University, Nanjing 210023, China)

**Abstract:** With the development of Internet technology, traditional WebGIS servers delay or even deny service when dealing with high concurrent access from large groups of users. After analyzing the problem, this paper proposes a framework of WebGIS servers based on cloud platform, which aims at dealing with high concurrent access. The framework uses cloud platform to provide flexible computing and storage resources for WebGIS servers, and alleviates bottleneck of high concurrency in three ways: load balancing, cache design, and database cluster. Open source server software GeoServer is selected to deploy as WebGIS application experiment. Experimental results show that usage of data cache can evidently reduce response time of WebGIS service. Compared with a single physical server, the WebGIS server clusters based on cloud platform can effectively process high-concurrency requests, and with the extending of cluster scale, the WebGIS system based on cloud platform can achieve good speedup.

**Key words:** cache; cloud platform; database cluster; high concurrency; load balancing; WebGIS

## 1 引言

近年来 Internet 的快速发展和 GIS 的广泛应用推动了 WebGIS 的诞生. WebGIS 是基于 Internet 平台采用 HTTP 协议的地理信息系统. GIS 专注于地理数据的获取、存储和管理, WebGIS 则提供给用户在任何地点通

过任何平台读取、发布、共享全球地理数据的服务<sup>[1]</sup>. 与传统 GIS 相比, WebGIS 的功能包括地图服务、地理编码、导航或位置服务, 它除了在国土、资源等传统领域应用外, 也促进了与人民生活息息相关的车载导航、移动位置服务、现代物流等产业的迅速发展.

<sup>①</sup> 基金项目:安徽省自然科学基金(1408085MKL06);国家自然科学基金(41371433);高等学校学科创新引智计划(B07033)

收稿时间:2016-03-02;收到修改稿时间:2016-04-14 [doi:10.15888/j.cnki.csa.005440]

随着 WebGIS 的普及,大用户群并发访问的情况愈发普遍,巨大的负载造成服务器性能下降,表现为响应时间变长、部分请求丢失,甚至出现超时拒绝服务的现象。所以 WebGIS 高并发瓶颈成为分布式 GIS 领域的研究重点。

目前针对该问题已有不少研究成果。算法方面英国爱丁堡大学 DTI/SERC 项目<sup>[2]</sup>分析了空间数据并行技术,开发了基于 NINID 的原型系统以及并行 GIS 函数库。空间数据方面殷小静<sup>[3]</sup>等人对地理信息存储采用瓦片金字塔建模,利用四叉树拓扑关系检索,从数据预装载、多线程等方面提高了渲染的实时性。系统架构方面工业界实现了 C/S 模式下 GIS 分布式组件模型,包括 DCOM 分布式对象模型组件、EJB 容器组件、.NET 分布式应用架构等,但组件模型有框架笨重、安全性不高、开发困难等缺陷。学术界中彭义春<sup>[4]</sup>等人分析了云 GIS 的四种应用模式和一个多层系架构模型,重点研究了虚拟化技术、分布式存储技术以及并行空间分析。Muzafa<sup>[5]</sup>等人从应用框架、成本维护和数据编码等方面阐述了云 GIS 的可行性。前两者从理论上研究了云 GIS 架构及应用模型,但缺乏具体的数据分析。万鹏<sup>[6]</sup>等人提出了基于 CloudStack 的 WebGIS 架构,并使用 ArcGIS 进行测试分析,但文中没有考虑到高并发访问对空间数据库 I/O 带来的瓶颈。

本文从系统架构角度出发提出了一种基于云平台的高并发 WebGIS 框架。云平台为 WebGIS 服务器集群提供弹性计算和存储资源。架构选用负载均衡器平衡系统负载,提高请求响应速度。空间数据缓存设计减少了数据库访问次数,降低通信代价。数据库集群用于缓解单点数据库在高并发时的 I/O 瓶颈。本文采用开源软件 GeoServer 作为 WebGIS 应用环境部署,实验表明缓存机制能显著降低服务响应时间。云平台 WebGIS 集群架构与单物理服务器相比能有效处理高并发请求,且随着集群规模扩大,WebGIS 系统获得了良好的加速比。本文研究成果对于分布式地理信息共享、GIS 社会化普及都有着重要意义。

本文剩余部分组织如下:第 2 节介绍云平台背景和分析,第 3 节介绍架构总设计以及高并发解决方案实施,第 4 节是实验结果及分析,第 5 节总结本文工作并指出下一步工作方向。

## 2 云平台背景分析

### 2.1 云平台概述

根据美国国家标准与技术学会定义:云计算是一种以较少的配置和交互方式使用户通过网络按需访问和配置计算资源池的模式<sup>[7]</sup>。云计算具有虚拟化、弹性扩展、快速部署等特征。云平台也称为云计算平台,平台允许开发者编写运行在云环境的应用程序或使用云服务。云平台服务分为三种模式:IaaS (Infrastructure as a Service,基础设施即服务)、PaaS (Platform as a Service,平台即服务)、SaaS (Software as a Service,软件即服务)。本文选用 IaaS 云平台部署,IaaS 模式的特色在于核心技术虚拟化将底层物理设备与上层操作系统、软件去耦合,多个物理设备纳入资源池统一管理,提高了资源利用率且有利于虚拟机的快速弹性扩展。

### 2.2 云平台 WebGIS 架构优势

云平台 WebGIS 是指 WebGIS 的平台、软件和地理信息以按需获取的方式,能够方便、高效部署到“云”基础设施之上,弹性地提供最广泛的 Web 服务<sup>[8]</sup>。传统的 WebGIS 分布式架构是基于 GIS 服务提供商的平台搭建,因此 WebGIS 系统在服务调用、数据存储等方面与底层技术耦合性较高。下面从三个方面分析云平台 WebGIS 架构相对于传统 WebGIS 架构的优势:

(1) 数据库访问。传统的 WebGIS 架构地理信息分布在不同的数据库中,导致了数据具有明显的异构性,降低了应用互操作性。云平台 WebGIS 将传统的数据库引擎统一为接口形式,屏蔽了地理数据多源异构的特点。

(2) 数据格式多样性。由于数据库异构导致地理信息格式多样化,传统的 WebGIS 架构频繁转化数据格式降低了系统性能。云平台 WebGIS 提供空间数据访问时使用 JSON 格式存储和传输地理信息,消除格式之间的互相转化。

(3) 系统扩展性。传统 WebGIS 动态添加实时性差,云平台 WebGIS 可以向资源池自动申请硬件资源,实现虚拟机服务器的快速部署和运行,资源池管理提高了资源利用率。

### 2.3 云平台 OpenStack 搭建

目前比较流行的开源 IaaS 云平台有 CloudStack、Eucalyptus、OpenNebula、OpenStack。与其它 IaaS 云平台相比,OpenStack 源代码完全开放、分层架构合理、部件耦合性低,借助 Libvirt 可支持多个虚拟机管理程序<sup>[9]</sup>,Aaron<sup>[10]</sup>等人通过实验数据证明 OpenStack 新建虚拟机耗时比其它平台如 CloudStack 更少,故本文采

用 OpenStack 搭建云平台。

OpenStack 是帮助服务商和企业内部实现基础架构服务的开源云计算管理平台,采用模块化设计,基本模块及功能如图 1 所示。其中 Nova 组件是 OpenStack 的核心,OpenStack 中虚拟机生命周期的所有活动都由 Nova 处理,它是负责管理计算资源、网络、认证、可扩展性的平台。

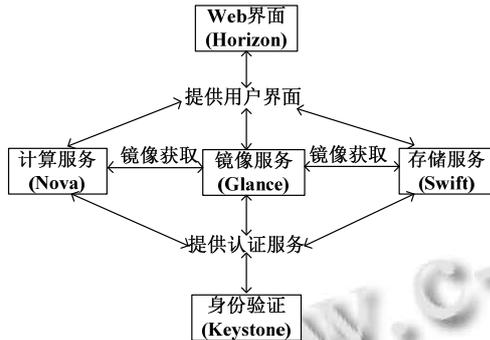


图 1 OpenStack 模块功能图

本文 OpenStack 云平台搭建在在六台物理服务器上,一台充当控制节点,其余为计算节点。每台服务器硬件配置除硬盘空间外基本相同,操作系统为 CentOS7,内核版本是 Linux version 3.10.0-229.el7.x86\_64。具体配置如下表 1 所示。

表 1 云平台物理服务器硬件配置表

配置项	配置值
CPU	Intel (R) Xeon (R) CPU E5-2620 v2 @ 2.10GHz
处理器数	24
内存	64GB
网卡带宽	千兆网卡

除了 Nova 外,其余服务都安装在控制节点,控制节点实现网络控制、调度管理等功能。Nova 在控制节点和计算节点均需安装。计算节点上 nova 组件提供 nova-compute 服务,控制运行在宿主机上的计算实例。Keystone 为所有服务提供身份认证,所以首先被安装。使用 yum 命令安装 openstack-keystone 等安装包,修改配置文件参数,完成后启动服务并创建用于服务验证的用户、租户、角色。其余组件安装过程大致相似。云平台安装完成后可通过 nova 命令或登陆用户界面上上传虚拟机操作系统镜像、制定硬件参数以及新建虚拟机操作等。

对于云平台而言,虚拟机的快速复制是一个重要机制。通常 OpenStack 使用虚拟磁盘快照模式复制,子虚拟机通过 qdisk 完成,每次复制都需要添加新的磁盘镜像。本文引用 SnowFlock API 进行虚拟机复制,它借鉴 UNIX 系统的 fork 语义,采用慵懒(lazy)的状态复制,按子虚拟机所需内存复制方式操作。SnowFlock 在随后过程中获取父虚拟机的状态信息,尽量减少状态转移,加快了虚拟机的横向扩展。

### 3 架构设计

本文从低成本可扩展性角度出发,选择从负载均衡、数据缓存和数据库集群三方面解决 WebGIS 高并发瓶颈,提出了一种基于云平台的 WebGIS 系统架构。该架构的目的是提高 WebGIS 面对高并发时的处理性能。架构如图 2 所示,从物理角度分为五层,分别是浏览器、负载均衡器、云平台、缓存数据库(包括 session 缓存和空间数据缓存)以及数据库集群。从逻辑上分为三层——表示层、逻辑层、数据层。

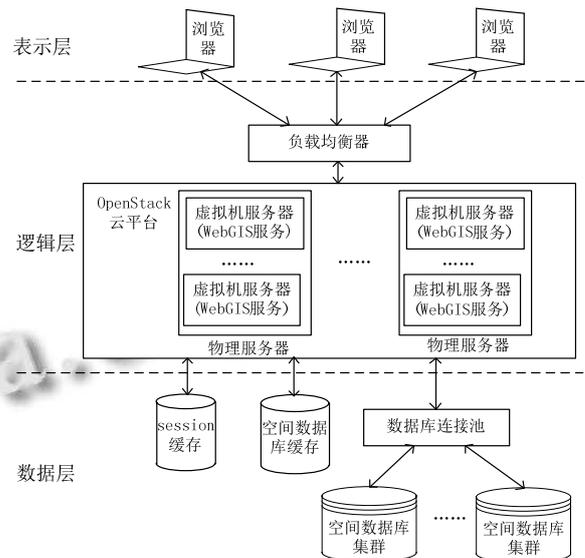


图 2 云平台 WebGIS 系统架构图

(1) 表示层。表示层浏览器的功能是提供用户与 WebGIS 系统交互,显示地图服务图像,并通过浏览器实现放大、缩小、漫游等视觉效果。

(2) 逻辑层。逻辑层包括负载均衡器和云平台。负载均衡器负责接收用户请求,并按照既定的负载均衡算法将请求分发到后台的云平台服务器中。OpenStack 云平台通过虚拟化技术为 WebGIS 服务器提供资源池。

云平台 WebGIS 虚拟机服务器负责解析数据请求,对数据业务并行处理,包括空间分析和空间查询等操作。

(3) 数据层. 数据层包括缓存数据库和空间数据库. 缓存 session 可实现服务器集群 session 共享. 空间数据缓存遵循既定读写机制,减少了数据库访问次数. 空间数据库集群存放 WebGIS 地理信息,数据库集群的部署方式缓解了单点 I/O 瓶颈,提高了数据库系统吞吐量。

当用户通过浏览器发出读取地理信息请求,负载均衡器接受请求并根据算法选择云平台 WebGIS 服务器进行转发,服务器解析请求,若该用户为第一次发送请求,则在内存数据库中为该用户创建 session 信息,若否则根据 session 验证身份,然后查看空间数据缓存中是否存在请求数据,存在则直接返回,不存在则读取数据库集群,返回数据后将读取的数据及周边信息写入缓存中。

本节剩余内容描述了云平台 WebGIS 架构中面对高并发采取的具体操作和算法,并使用 GeoServer 作为 WebGIS 应用示例部署于架构中。

### 3.1 负载均衡

#### 3.1.1 负载均衡原理

负载均衡技术将来自用户的高并发请求通过使用某种负载均衡算法分发至集群中提供同一服务的多个服务器节点<sup>[11]</sup>,负载均衡器把多台服务器的地址映射为一个统一的对外服务 IP 地址,该操作对用户透明. 负载均衡器平衡了系统负载,提高了服务器响应速度. 当某台服务器发生故障时,用户请求仍可通过负载均衡分发到正常工作的服务器上,保障了网站的高可用性。

#### 3.1.2 负载均衡器 Nginx

Nginx 作为一个高性能的 HTTP 和负载均衡服务器,使用操作系统提供的 I/O 多路复用技术(通过 epoll 系统函数调用),消除了向内核传递连接和内核遍历连接等耗时操作. Nginx 因其出色的处理高并发的性能以及轻便简洁的经济特性适用于中小型网站<sup>[12]</sup>. 相比较 Apache, Nginx 异步非阻塞式处理请求,能在高并发下保持低资源消耗. 所以本文选择 Nginx 为云平台 WebGIS 系统的软件负载均衡器. 配置文件中 work\_processes 指令描述 Nginx 创建的单线程数目,调整该值至服务器 CPU 总核数可获得性能提升. 开启 gzip 压缩加快数据传输,由于压缩会带来额外负载,所以避免压缩小型文件如图片、PDF 等。

#### 3.1.3 负载均衡算法

Nginx 的 upstream 扩展模块是实现负载均衡器的具体功能模块,它通过回调函数向后端服务器读写数据. 模块中默认有三种负载均衡算法: 轮询、ip-hash、least-connected. 轮询依据到达顺序分发请求,没有考虑后台服务器使用情况,且无法实现 session 共享. ip-hash 解决了 session 共享问题,但要求 Nginx 为最前端服务器否则得不到正确的 IP 地址,而且易造成分配不均. least-connected 算法遍历后端服务器集群,选取 connctions/weight 数值最小的服务器分发请求,可以根据后台服务器情况反馈请求分发机制,所以本文选择 least-connected 作为 Nginx 负载均衡算法。

### 3.2 数据缓存

#### 3.2.1 缓存数据库 Redis

数据缓存包括 session 存储和空间数据缓存. 相对于磁盘,内存数据库的读写速度要高出几个数量级,数据保存在内存中访问更能获得优越的应用性能. 常见的内存数据库包括 FastDB、Memcached 以及 Redis. Redis 因其持久化功能、丰富的数据类型、高稳定性等优势满足本文缓存需求,且 Redis 的 key-value 数据结构适合 sessionID-userID 的存储。

Redis 有两种持久化方法, Snapshot 以及 AOF(Append Only File),由于 session 共享对于数据丢失面临的影响并不显著,且主从模式冗余存储信息备份,所以选择关闭持久化功能以提高读写性能. Redis 的 maxmemory 表示数据库能使用的最大内存值,通过调整 maxmemory 数值并使用 redis-benchmark 命令测试 Redis 读写性能,得到如图 3 所示的结果. 从图中可以看出当 maxmemory 达到物理内存 60%之前,读写性能呈递增趋势,从 60%增加到 80%时,SET 和 GET 操作性能增长速度明显变缓慢,所以认为设置 maxmemory 设为物理内存的 60%到 80%区间较为合理。

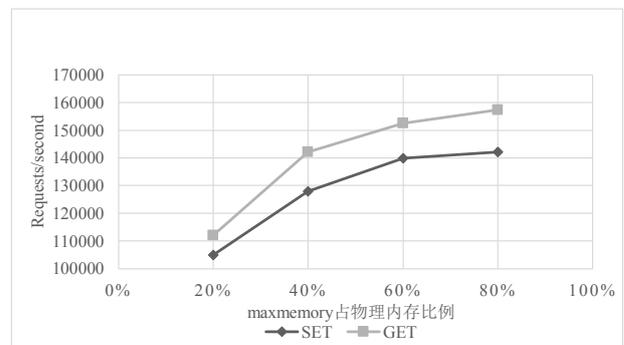


图 3 Redis 读写性能测试图

### 3.2.2 session 共享

Redis session 共享配置在应用服务器端添加, 根据服务器类型不同配置方式不同, 如 tomcat 服务器使用 sessionmanager 方法存储 Redis 服务器的地址、端口号等信息. 当 Redis 内存溢出时, 选择 allkeys-lru 算法替换, 即标记长时间未使用的 session 为过期状态.

### 3.2.3 空间数据缓存

Redis 内存数据库作为空间数据缓存设备, 提高了数据访问速度, 并通过缓存有效减少数据库访问次数, 降低了通信代价. 本文设计并实现了 Redis 与空间数据库之间的地理信息缓存读写机制, 缓存读写机制流程如图 4 所示.

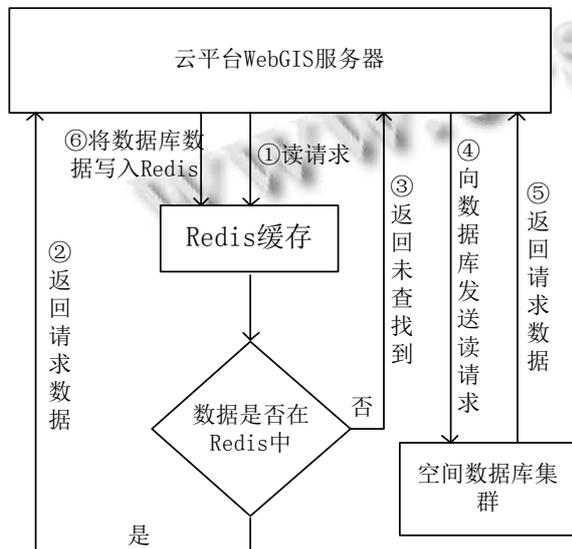


图 4 Redis 空间数据缓存读写机制流程图

首先云平台 WebGIS 服务器向 Redis 缓存数据库发出读请求, 判断数据存在则直接返回请求结果, 若不存在则返回服务器未查找到数据, 服务器根据返回信息向数据库发送读请求, 待数据库请求返回后依照局部访问原理将该数据所在周边元素写入到 Redis 中.

数据库缓存的写请求机制相对简单, 服务器直接修改空间数据库信息, 如果该信息之前存在于 Redis 之中, 则将其删除.

本文选择 OSM (OpenStreetMap) 数据格式在 GeoServer 中发布 WMS 服务. 从 OSM 官网上可以导出国家和城市的地理矢量数据, OSM 导入数据库中包含不同的表和视图. 以表 planet\_osm\_line 为例, 表中一条记录包含了众多属性, 包括主键 osm\_id、way\_area 折线区域、railway 是否为铁路等 68 个, 如下表 2 所示.

表 2 OSM 数据库 planet\_osm\_line 属性表

osm_id	.....	railway	.....	way_area	.....
--------	-------	---------	-------	----------	-------

要将数据库中一条记录转化为 Redis 的 key-value 结构, 需要选取合适的数据类型. Redis 中的 hash 类型符合要求. Redis hash 是一个 string 类型的 field 和 value 映射表, 一个 key 可以对应多个 field, 每个 field 对应一个 value. Hash 类型的 key 值设计为主键, 每个 field 对应于各属性名, value 对应于属性值. 数据转化如下表 3 所示.

表 3 OSM 数据 hash 格式转化表

key	field1	value1	field2	value2	...
osm_id	railway	0	way_area	01...B1	...

要在 GeoServer 源码中添加空间数据缓存功能, 首先需要使用 Maven 将 GeoServer 源码导入 Eclipse 建立工程. GeoServer 采用 Wicket 开发框架, 并使用了近百个第三方软件包, 如 GeoTools 是 Java 语言编写的开源 GIS 工具包.

GeoServer 源码中为数据库连接创建了一个类 JDBCConnectionPoolPanel, 该类为数据库连接池制定一个模板, 本文在该类中添加缓存读写机制代码, 核心代码如下图 5 所示.

```

1. public WmsLayer getLayer(int osm_id) //读机制核心代码
2. {
3.     String key=RedisKey(osm_id); //定义访问数据的key
4.     boolean exist =RedisUtil.isexist(key);
5.     //查看该key是否存储在redis中
6.     if (exist)
7.     { //如果存在则返回, 将JSON字符串转化为Java对象
8.         return JsonUtil.fromJson(RedisUtil.get(key), WmsLayer.class);
9.     }else{
10.        WmsLayer wmslayer=new WmsLayer();
11.        wmslayer.getDBdata(osm_id);
12.        //根据osm_id读取数据库中的数据
13.        RedisUtil.set(key, JsonUtil.toJson(wmslayer));
14.        //将数据库读取的java对象数据转化为JSON格式存入Redis
15.        return wmslayer;
16.    }
17.    .....
18.}
19. public void writeLayer(WmsLayer wmslayer)//写机制核心代码
20. {
21.     wmslayer.writeDBdata();//写数据库
22.     String key=RedisKey(wmslayer.get(osm_id));
23.     //定义写数据的key
24.     boolean exist =RedisUtils.isexist(key);
25.     //判断该key是否在redis中
26.     if(exist)
27.         jedis.del(key);//存在则删除
28.}
    
```

图 5 Redis 缓存读写机制关键代码

### 3.3 空间数据库集群

#### 3.3.1 数据库集群优势

数据库集群将多台数据库服务器构建一个虚拟单一的数据库逻辑映像,根据客户端信息或者请求类型等选择登陆集群的数据库进行并行查询,其目的是突破单点数据库服务器 I/O 瓶颈,解决数据库扩展性问题,提高 web 应用系统性能。

#### 3.3.2 PostGIS 集群实现与优化

PostGIS 是对象-关系型数据库 PostgreSQL 的扩展,提供以下空间信息服务功能:数据库坐标变换、栅格数据类型、空间聚集函数等<sup>[13]</sup>。与 Oracle 的 Spatial 数据库相比,PostGIS 具有更好的自动内存管理功能和页面替换策略。

本文采用 Share-Nothing 模式搭建数据库集群,集群中的每个节点保持完整的数据集,每台数据库服务器都能提供服务,提高了数据库吞吐量。数据分片采用复制方式,这种方式数据一致性维护代价较大,通常采用异步复制更新数据库。

在 PostgreSQL 安装过程中通过调整配置文件参数可以使性能得到很大提升。根据文献[14]中使用自动参数设置工具 iTuned 测试 shared\_buffers 以及 effective\_cache\_size 的变化对数据库性能影响,当 shared\_buffers 超过物理内存的 20%时增加缓冲池反而会削弱数据库性能,这是由于 shared\_buffers 增加导致操作系统文件缓存的内存减少,因此磁盘 I/O 溢出分区增加,继而性能下降。Effective\_cache\_size 达到物理内存的 30%时任务执行时间最短,因为优化器更倾向于使用索引扫描替代顺序扫描。配置文件中 Fsync 参数强制把数据同步到磁盘,设置为 off 可缓解系统 I/O 压力。

此外,架构利用 pgpool-II 中间件连接数据库集群和客户端服务器,它提供数据库复制、连接池功能,对于多用户的并发查询进行负载均衡,并设置并行查询降低总体执行时间。

## 4 实验测试与分析

为了验证云平台 WebGIS 架构面对高并发时的处理能力,本文通过 GeoServer 应用实验来对比分析。

GeoServer 是 Java 开发的开源 WebGIS 服务器软件,允许用户共享和编辑地理空间数据。它符合开放地理信息系统协会规定的 WFS 和 WCS 标准,并提供高性能认证的 WMS 服务<sup>[15]</sup>。实验测试页面是 WMS

服务显示图,如图 6 所示。WMS 根据用户查询条件获取空间要素并进行地图绘制和渲染,是典型的计算密集型服务。在高并发下处理 WMS 服务请求能够较好反映 GeoServer 服务器的处理能力<sup>[16]</sup>。服务数据采用中国局部范围的 OSM 数据格式(数据源本身不包含南沙群岛周边十段线)。

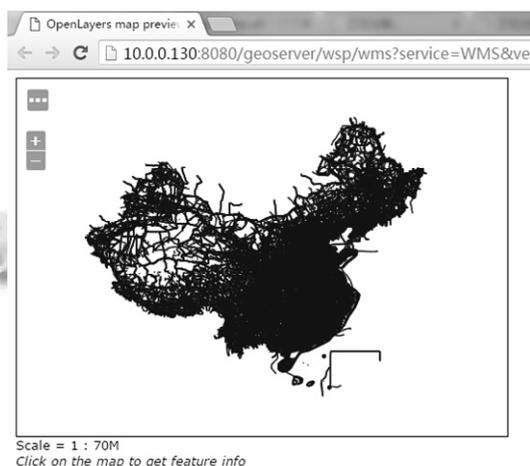


图 6 GeoServer WMS 显示图

测试工具使用 Jmeter2.13。实验中 Jmeter 的测试计划除了线程数(并发用户数)不同外,启动时间均为 0 秒,表示同时启动,循环次数为 1 次。

#### 4.1 缓存机制性能测试

缓存机制性能测试将 4 台相同的 GeoServer 虚拟机服务器组建集群,并在 Nginx 的 upstream 模块中配置虚拟机信息和 least-connected 算法, Jmeter 脚本并发访问 GeoServer 发布的 WMS 服务数据。虚拟机服务器的硬件配置如下表 4 所示。

表 4 GeoServer 云平台虚拟机服务器硬件配置

配置项	配置值
处理器数	4
内存容量	4GB
硬盘容量	20GB

实验对比了官网源代码与添加缓存功能代码在高并发访问 WMS 动态页面时的平均响应时间,表 5 是测试结果。

表 5 缓存性能测试数据(单位: 毫秒)

并发数	100	200	300	400	500
无缓存	4169	6642	11962	20359	26131
使用缓存	2877	4105	6688	11808	17102

从表中可以看出使用缓存机制明显缩短了响应时间. 当并发数较小时, 响应时间缩短比例为 30%左右, 随着并发数逐渐增加, 缓存数据库中已经保存了之前访问的空间数据, 数据命中率提高, 当并发数为 300 时缓存机制效率最高, 时间相比无缓存代码缩短了 45%. 而随着并发数的进一步增加, 数据替换导致命中率有所下降, 所以时间缩短的比例有所降低.

#### 4.2 云平台服务器高并发测试

为了验证云平台 GeoServer 架构在面对高并发时的处理能力, 本文通过测试不同虚拟机个数在线程数递增时的平均响应时间与单个物理服务器响应时间作对比. 虚拟机配置与表 4 相同, 单个物理服务器硬件配置如表 6 所示.

表 6 单物理服务器硬件配置表

硬件配置项	配置值
CPU	AMD Athelon(tm) X4 750 Quad Core Processor 3.40 GHz
处理器数	4
内存	4GB
网卡带宽	千兆网卡

测试脚本访问 GeoServer 发布的 WMS 服务, 实验测试了云平台服务器集群节点个数为 2/4/6/8 时的情况. 随着并发数逐渐增加, 服务平均响应时间如下图 7 所示.

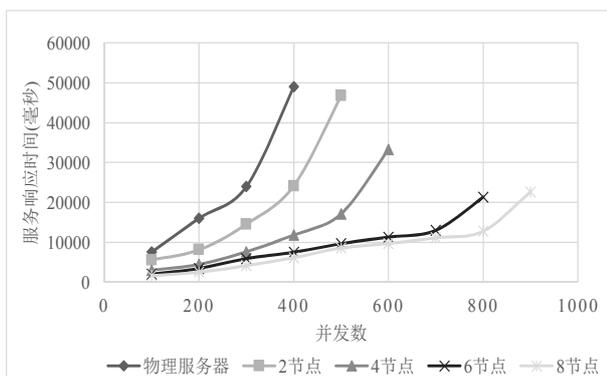


图 7 GeoServer 服务器集群与物理服务器性能比较

从图 7 中可以看出物理服务器在并发数为 400 时反应时间大幅增加, 出现高并发瓶颈. 相同的并发条件下, 云平台服务器集群的响应时间均小于物理服务器. 且随着云平台 GeoServer 集群节点数增加, 系统能处理的并发量上限越来越高.

云平台中虚拟机的扩展性是一个重要特性. 加速

比是反映扩展性的一个衡量标准. 本文使用单台虚拟机服务器的响应时间与不同规模集群的响应时间相除计算加速比. 在并发量 300 时测试数据得到的加速比曲线如图 8 所示, 从图中可看出由于服务器集群需要额外开销, 实际加速比低于理想值, 但总体来说云平台 WebGIS 系统能够获得良好的加速比.

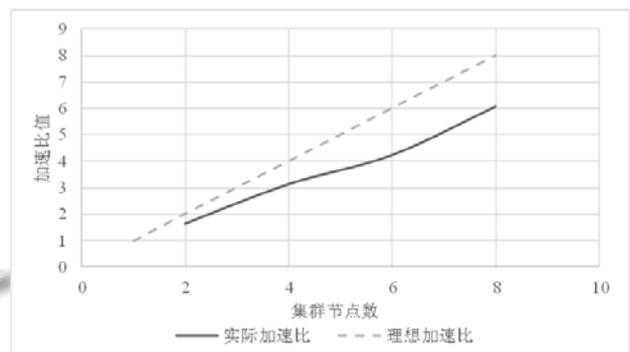


图 8 并发量 300 时加速比曲线

## 5 总结

本文针对 WebGIS 服务器在面对高并发请求时出现的性能瓶颈提出了一种基于云平台的 WebGIS 服务器架构, 并利用负载均衡、空间数据缓存等方案改善了架构的高并发性能. 文中采用 GeoServer 软件作为 WebGIS 应用环境设计对比实验. 实验结果表明缓存机制的使用能够有效降低 WebGIS 服务响应时间; 相对于单物理服务器而言云平台能更好地处理高并行请求, 且随着云平台服务器集群规模增加, WebGIS 系统能获得良好的加速比. 后续工作将继续对架构改进, 如研发基于 Nginx 动态反馈负载均衡算法, 基于 auto-scaling 实现 WebGIS 虚拟机的自动伸缩扩展等, 以期进一步提高云平台 WebGIS 服务器的高并发性能.

### 参考文献

- 1 Liu XH, Han JZ, Zhong YQ, Han CD, He XB. Implementing WebGIS on Hadoop: A case study of improving small file I/O performance on HDFS. Proc. of the 2009 IEEE Conf. on Cluster Computing and Workshops. 2009. 1-8.
- 2 Richard GH, Steve D, Bruce MG, Mike M. Parallel processing algorithms for GIS. London: Taylor and Francis, 1998: 13-45.
- 3 殷小静, 慕晓冬, 徐义文, 陈琦. 海量地形数据的管理和交互策略优化. 计算机应用, 2011, 31(9): 2465-2467.
- 4 彭义春, 王云鹏. 云 GIS 及其关键技术. 计算机系统应用, System Construction 系统建设 67

- 2014,23(8):10-17.
- 5 Muzafar AB, Razeef MS, Bashir A. Cloud computing: A solution to Geographical Information Systems (GIS). *International Journal on Computer Science and Engineering*, 2011, 3(2): 594-600.
- 6 万鹏,黄方,赵伟彪,卜栓栓,董蕾,冯杰.基于云计算的并发访问密集型 WebGIS QoS 研究. *地理信息世界*,2013,20(4): 20-26.
- 7 Mell P, Grance T. The NIST definition of cloud computing. National Institute of Standards and Technology, 2011.
- 8 王凤领.基于云计算的 WebGIS 分析架构研究. *计算机技术与发展*,2014,24(3):113-117.
- 9 Anton B, Sareh FP, Mohammed A, Rajkumar B. Deploying OpenStack on CentOS using the KVM hypervisor and GlusterFS distributed file system [Technical Report]. CLOUDS-TR-2012-3. Cloud Computing and Distributed Systems Laboratory, University of Melbourne, 2012.
- 10 Aaron P, Liu L, Yuan B. Bench marking the performance of OpenStack and CloudStack. *Proc. of the 2014 IEEE 17th International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing (ISORC)*. 2014. 405-412.
- 11 梁啟成.面向公有云存储的高并发关键技术研究及系统实现[硕士学位论文].广州:华南理工大学,2013.
- 12 凌质亿,刘哲星,曹蕾.高并发环境下 Apache 与 Nginx 的 I/O 性能比较. *计算机系统应用*,2013,22(6):204-208.
- 13 张大鹏,张锦,郭敏泰,李海涛.开源 WebGIS 软件应用开发技术和方法研究. *测绘科学*,2011,36(5):193-196.
- 14 Duan SY, Vamsidhar T, Shivnath B. Tuning databases configuration parameters with Ituned. *Proc. of the VLDB Endowment*, 2009, 2(1): 1246-1257.
- 15 Fang Y, Min F. A WebGIS framework for vector geospatial data sharing based on open source projects. *Proc. of the International Symposium on Web Information Systems and Applications*. 2009. 124-127.
- 16 范协裕,任应超,杨崇俊,唐建智,易雄鹰.基于集群技术的可伸缩云 GIS 服务平台研究. *计算机应用研究*,2012,29(10): 3736-3739.