

文章编号:1673-8411(2019)01-0074-05

基于 OpenStack Swift 的雷达产品存储研究

王丽玫, 李莹, 黄志, 马军

(广西区气象信息中心, 广西 南宁 530022)

摘要: 简要介绍分布式对象存储系统 OpenStack Swift, 包括安装部署及其在雷达产品存储方面的应用, 研究证明, OpenStack Swift 能较好的支持雷达产品存储, 也适用于其它气象资料存储管理。

关键词: 分布式; 雷达产品; 对象存储

中图分类号: P412.25

文献标识码: A

Research on Radar Product Storage Based on OpenStack Swift

Wang Limei, Li Ying, Huang Zhi, Ma Jun

(Guangxi Meteorological Information Center, Nanning Guangxi 530022)

Abstract: This paper briefly introduces OpenStack Swift, a distributed object storage system, including its installation and deployment, and its application in radar product storage. The research shows that OpenStack Swift can better support radar product storage and is also suitable for other meteorological data storage management.

Keywords: distributed; radar products; object storage

1 引言

广西新一代天气雷达观测投入应用以来, 生成的观测和产品数据迅猛增长, 对数据的存储和管理提出了较高的要求。传统的存储由于成本昂贵、性能扩充有限, 无法满足气象业务对数据存储和管理的需求。针对海量数据的存储和管理, 一般解决方案为构建大规模廉价服务器集群, 采用分布式存储技术, 通过添加节点的方式动态伸缩以保证可扩展性。分布式存储可以分为三类: 块存储、文件存储、对象存储, 开源项目有 HDFS、Ceph、Swift、ClusterFS 等, 商业实现有 Google、AWS 等。Swift 是一个多租户、高扩展性和高可用性的对象存储系统, 用于低成本地存储大量的非结构化数据^[1]。目前对 Swift 有较多的试验和应用, 如王胜俊等^[2]开展了“Openstack 云平台中 Swift 组件的研究与测试”, 郑驰等^[3]开展了“基于 OpenStack 的对象存储性能实验及研究”, 谢超群^[4]已经实现了“基于 Swift 云存

储技术在高校数据中心的应用”, 李磊等^[5]建立了“基于 OpenStack Swift 构建高可用私有云平台”; 在气象应用方面, 刘光博^[6]开展了“分布式文件系统在气象业务中的应用初探”。本文则尝试将 Openstack Swift 对象存储系统应用于气象资料的存储管理, 解决大批量雷达产品存储管理和应用的问题。

2 openstack swift 简介

OpenStack 起源于 Rackspace 和美国航空航天局 (NASA) 共同开发的云计算平台, 用于帮助服务商和企业内部实现类似于 Amazon EC2 和 S3 的云基础架构服务。OpenStack 开始时包含两个主要模块: Nova 和 Swift。OpenStack 作为集中大量计算、存储和网络资源、提供大规模可扩展的云操作系统, 具备了模块松耦合、虚拟化技术全面、组件配置灵活、便于二次开发的优势。

Swift 搭建在标准硬件存储基础设施之上, 通过在软件层面引入一致性散列技术和数据冗余

收稿日期: 2019-01-10

基金项目: 广西区气象局气象科研项目“基于 OpenStack Swift 的分布式气象雷达产品存储研究”(项目编号: 桂气科2017M06)资助。
作者简介: 王丽玫 (1976-), 女, 广西玉林人, 大学本科, 高工, 主要从事气象信息网络工作, E-mail:924639347@qq.com。

表 1 Swift 支持的操作

资源类型	URL	GET	PUT	POST	DELETE	HEAD
账户	/account/	获取容器列表	-	-	-	获取账户元数据
容器	/account/container	获取对象列表	创建容器	更新容器元数据	删除容器	获取容器元数据
对象	/account/container/object	对象内容和元数据	创建、更新或拷贝对象	更新对象元数据	删除对象	获取对象元数据

性, 牺牲一定程度的数据一致性来达到高可用性和可伸缩性。Swift支持多租户模式、容器和对象读写操作, 适合解决互联网的应用场景下非结构化数据存储问题。Swift存储的层次结构为租户/容器/对象, 租户可以创建多个容器, 容器中创建多个对象。

Swift 通过 Proxy Server 向外提供基于 HTTP 的 REST 服务接口, 对账户、容器和对象进行 CRUD 等操作。在访问 Swift 服务之前, 需要先通过认证服务获取访问令牌, 然后在发送的请求中加入头部信息 X-Auth-Token。Swift 支持的操作可以总结为表 1。

3 Swift 部署

系统全部在虚拟机环境下搭建, 由 1 台控

制节点(含代理服务、认证服务)和 40 台存储节点组成, 硬件配置分别为 16vCPU/16GB 内存 /100GB 硬盘 / 网卡 1, 以及 4vCPU/4GB 内存 / 硬盘 60 (系统)+100GB (存储) / 网卡 1。OpenStack Swift 使用 Pike 版本, 操作系统选用 SUSE Linux 12 SP3。图 1 为系统逻辑结构图。

3.1 控制(代理)节点安装

控制节点安装包括网络配置、时钟同步设置、OpenStack 包、数据库、消息队列、内存对象缓存、身份识别、代理服务等, 安装顺序见图 2。其中 Dashboard 是一个网页版的交互式管理工具。

3.2 存储节点安装

控制节点安装包括网络配置、时钟同步设置、支持包、创建文件系统、swift 组件安装配置等, 安装顺序见图 3。

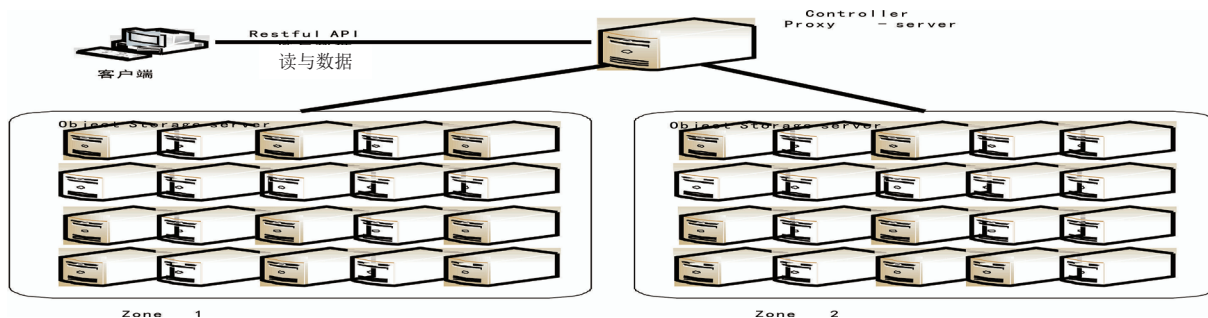


图 1 系统逻辑结构图

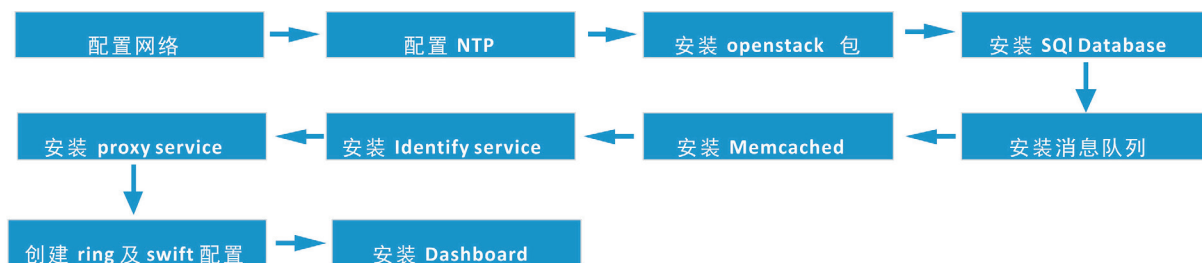


图 2 控制及代理节点安装步骤图



图 3 存储节点安装步骤图

3.3 安装完成验证

安装完成后可以在控制节点开展验证操作:

```
#source /etc/keystone/demo-openrc      -- 设置账户环境
#swift stat                             -- 查看 swift 信息
#openstack container create container1   -- 创建测试容器
#vim aa.txt
#openstack object create container1 aa.txt -- 容器写入对象
#openstack object list container1        -- 列出容器对象
#openstack object save container1 aa.txt  -- 读出对象
#openstack object delete container1 aa.txt -- 删除对象
#openstack container delete container1    -- 删除容器
```

上述操作能成功执行说明 swift 已经成功安装。

4 客户端应用

Swift 支持多种形式的客户端程序, 包括命令行界面 (CLI)、图形用户界面 (GUI)、基于网络的软件等。以下为系统中使用的两种客户端。

4.1 cURL

客户端 URL (cURL) 是一个预装在操作系统中的命令行工具, 通过它可以使 URL 语法和服务端传输数据^[1]。对 HTTP 请求提供了完善的控制, 可以处理所有的 Swift 请求。cURL 请求包括以下内容

- 1) curl 命令
- 2) -X <method> 选项提供 HTTP 动词 (如 GET、PUT)
- 3) 身份验证信息
- 4) 存储 URL
- 5) 数据和元数据 (可选)

如列出帐户的所有容器:

```
curl -X GET [...] http://swift.example.com/v1/AUTH_bob
```

4.2 开源客户端库 Jclouds

Apache Jclouds 是 Java 平台的开源多云工具

包, 它使用户可自由地创建跨云的应用程序, 同时为用户提供使用云特定功能的完全控制。该 API 提供云计算环境可移植抽象层以及云规范特性, 支持包括 Amazon, VMWare, Azure, OpenStack and Rackspace 等云计算平台。其中针对 OpenStack, 目前支持 Nova、cinder、glance、Neutron、Keystone、swift 等 API 接口。Jclouds 针对 swift 常用的类见表 2: 通过上述类, 可以满足对 openstack swift 的账户、容器和对象操作。

5 数据存储

系统处理的数据对象包括广西连续三年 (2014 ~ 2016 年) 全部新一代雷达产品, 文件总量约为 7000 万个, 每个观测站日文件产生量约为 2000 ~ 9000, 月文件产生量约为 6 万 ~ 27 万。如图 4 所示, Swift 中建立项目 demo, demo 下创建两种类型的容器, 分别存储逐站逐月、逐站逐日的产品文件, 以检验两种方式下文件存取效率差异。数据写入以 shell 脚本来实现, 可同时提交 20 个以上的脚本运行, 根据 shell 脚本的输入参数确定处理的站号和时间段。脚本的核心命令为:

表 2 Swift 常用的类及方法

序号	类	方法
1	AccountApi	deleteMetadata get updateMetadata updateTemporaryUrlKey
2	ContainerApi	create deleteMetadata get list update updateMetadata
3	ObjectApi	copy copyAppendMetadata delete deleteMetadata get getWithoutBody list put updateHeaders updateMetadata

```
#openstack container create ${containerName}
#openstack object create ${containerName} $radaPupName
```

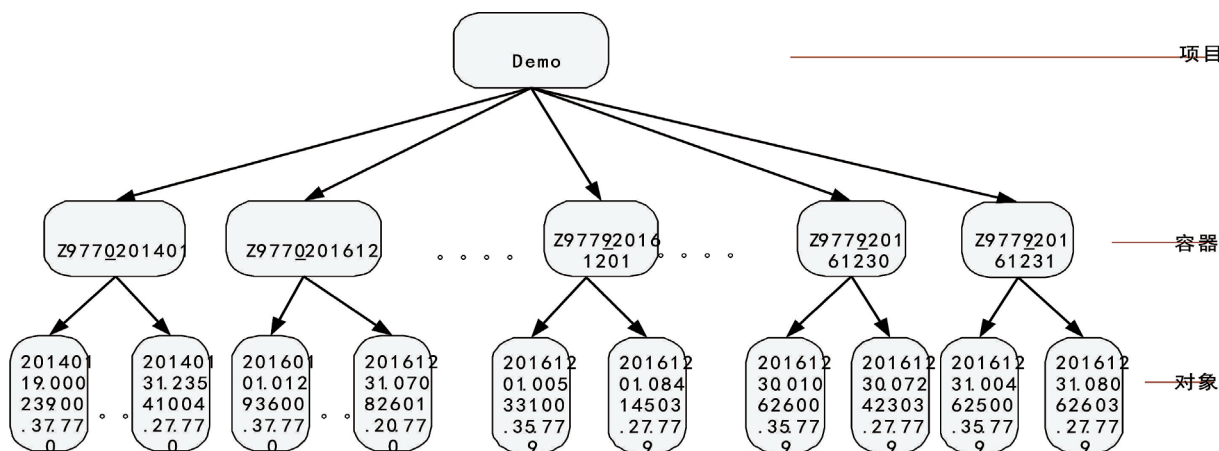


图4 数据存储逻辑结构

6 性能测试

6.1 写性能测试

使用6个进程同时对产品文件顺序写入，分别记录每次写入时长，经过分析发现，最小耗时为3s，最大耗时46s，以耗时4s内的记录居多，占95%以上。各时段数量分布见表3。

6.2 读性能测试

使用多台虚拟机模拟多个并发用户随机读取某个对象并下载到本地，记录每次操作耗时，得到的统计数据见表4。

6.3 测试期间资源使用情况

测试期间对控制（代理）节点、存储节点和客户端的主要资源使用情况进行观察，总体情况见表5。

7 分析

由于系统部署在虚拟化环境，全部虚拟机集

中部署在相同的数据存储中，并未获得理想的IO效果，写和读操作最小耗时基本为4s左右；随着并发访问量的增加，存储系统的平均响应时间也随之提高，但最大响应时间没有显著变化。另一方面，数据的读取性能还受制于客户端使用的并发线程数，并发数少的客户端平均读取数据用时更短。测试期间从资源使用情况来看，各节点的资源占用率较低，不对平台的读、写性能产生影响。

考虑改进的方向有：1) 使用配置多个硬盘的物理机搭建存储节点；2) 采用多个proxy-server+负载均衡提升响应能力^[3]；3) 前、后端网络分开，后端网络带宽压力是前端的至少3倍；4) 对象较多的时候采用多个容器分散存储，可以比存储在单个容器中获得更低的响应时间。

表3 写入耗时数量分布统计表

用时 (S)	占比 (%)
3	20%
4	75.74%
5 ~ 10	3.66%
>11	0.53

表4 不同并发数访问耗时统计表

并发数	使用虚拟机数量	使小用时 (秒)	使大用时 (秒)	平均用时 (秒)
50	1	4	46	12.52
200	4	4	52	28.84
210	7	4	34	17.84

表5 不同并发数访问耗时统计表

节点类型	CPU 最高使用率 (%)	内存最高使用率 (%)
控制节点	24	18
存储节点	16	22
客户机	5	25

8 结语

采用 openStack Swift 实现雷达产品存储管理是构建气象数据中心海量文件存储池的一次有益尝试, 随着气象业务和科研的不断发展, 除了雷达产品之外, 区域站、土壤水分、农业气象等资料越来越多, Swift 在高可用性、冗余性和高吞吐量方面的优势, 使得它成为较好的一个选择。

参考文献:

- [1] Joe Arnold, SwiftStack 开发团队. 对象存储: OpenStack Swift 应用、管理与开发[M]. 中国工信出版集团, 2017: 40-42.
- [2] 王胜俊, 李秦伟, 刘志聪. Openstack 云平台中 Swift 组件的研究与测试[J]. 电脑知识与技术, 2016, 12(7): 77-78.
- [3] 郑驰, 赵建军, 李成金, 等. 基于 OpenStack 的对象存储性能实验及研究[J]. 微型机与应用, 2014, 33(18): 15-16.
- [4] 谢超群. 基于 Swift 云存储技术在高校数据中心的应用[J]. 信息技术, 2016, (7): 158-160.
- [5] 李磊, 李达港, 金连文, 等. 基于 OpenStack Swift 构建高可用私有云存储平台[J]. 实验技术与管理, 2015, 32(5): 141-145.
- [6] 刘光博. 分布式文件系统在气象业务中的应用初探[J]. 气象研究与应用, 2009, 30(1): 88-90, 100.