

**Design of Distributed Object Cloud Storage Platform and Development of Third-party Application**

A Dissertation Submitted for the Degree of Master

**Candidate：Lai Wenhua**

**Supervisor：Prof. Wu Zongze**

SouthChinaUniversity of Technology

Guangzhou, China

**分类号： TP393 学校代号：10561**

**学号：201121009372**

华南理工大学硕士学位论文

分布式对象云存储平台设计

及第三方应用开发

作者姓名：赖文华 指导教师姓名、职称：吴宗泽 副教授

申请学位级别：全日制工程硕士 学科专业名称：电子与通信工程

研究方向：通信理论与网络技术

论文形式：□产品研发 □工程设计 应用研究 □工程/项目管理 □调研报告

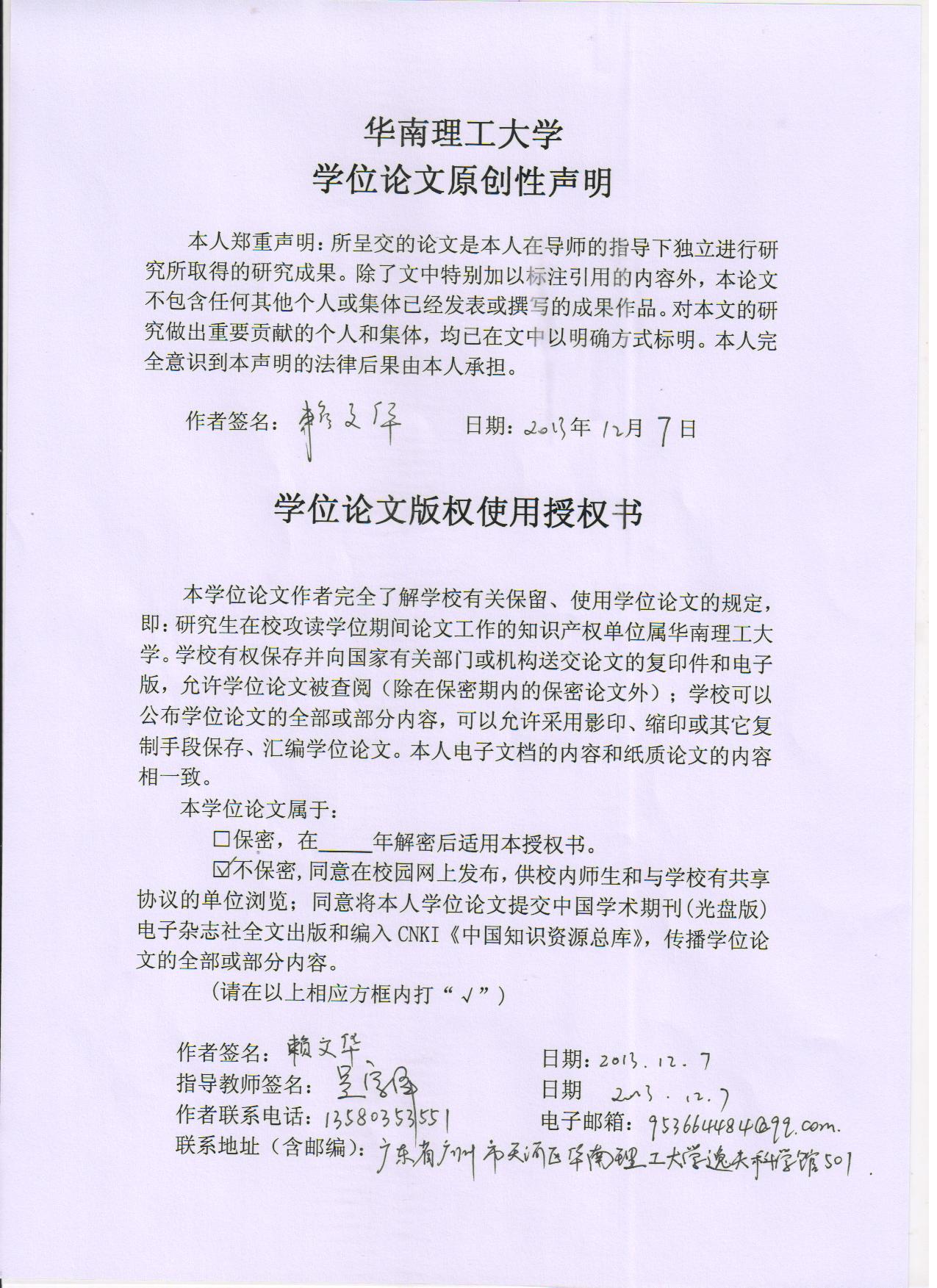
论文提交日期： 2013年10月23日 论文答辩日期： 2013年12月7 日

学位授予单位：华南理工大学 学位授予日期： 年 月 日

答辩委员会成员：

主席： 周智恒 副教授

委员： 傅予力 教授 吴宗泽 副教授 李波 副教授 李宗恒 高工

****

摘 要

互联网及其应用的发展带来的企业和个人数据量呈现出爆炸式增长，传统方案的计算、存储能力已经不能适应互联网高速发展的需求，云计算技术成为解决这一问题的革命性技术手段。中国电信作为国内三大运营商，面对国内正在发展庞大的云服务市场，更加需要整合自身丰富的带宽资源和庞大的企业及公众用户数量资源，对企业内部有进行海量的客户数据存储、管理以及业务分析的需要；对外可以提供存储服务接口提供高可用、可弹性拓展的云存储服务。

首先，本文对国外云存储服务提供商进行调研，深入分析国外云存储服务的技术发展状况和国外云存储产品的具体情况，同时对传统的存储技术进行剖析，分析它们的特点与不足，提出一个适合企业级的分布式对象云存储系统部署方案。

其次，本文利用开源代码项目Openstack，提出一个可行的云存储服务系统框架，设计并实现一个提供基本存储能力的内容管理云存储平台和业务管理的应用管理平台。内容管理平台是基本的云存储服务模块，负责基本存储服务功能(如创建容器、文件上传下载等)；应用管理平台是提供给服务提供商管理员进行日常维护、统计分析、费用结算等业务功能。

再次，设计并开发第三方应用接口，将本文的云存储系统做为WEB服务器的后台数据存储平台，WEB应用服务器利用本系统设计的第三方应用开放接口与云存储系统进行交互，实现以HTTP方式进行多媒体文件的存储服务。

最后，本文还对云存储系统的各个功能模块进行功能测试，测试表明本文云存储系统不仅拥有良好的存储服务功能，还提供适合的业务运营和系统运维功能。另外，第三方应用接口的开发，使用云存储系统做为WEB服务器的后台数据存储平台，有效的将数据存储和应用请求分离开来，解决了传统WEB应用部署中服务器因请求过大造成响应延迟问题，并从另一方面也验证了本文云存储系统的可用性。

关键字：分布式存储，对象存储，Openstack，云存储

# 

# Abstract

As a result of the development of the Internet and application, the data of enterprise and person has shown explosive growth trends. Traditional scheme of computing and storage capacity can’t adapt to the needs of the development of the Internet high-speed cloud computing, and in consequence storage become a revolutionary technology to solve this problem.Therefore, China telecom, as one of domestic telecom operators,need to integrate their own abundant bandwidth resources and user resources which could meet the demand of the storage, management and business analysis of the vast amounts of customer datawithin their businesses and could provide storage service interface to provide elastic expanding cloud storage service.

First of all,this paper researched the cloud storage service providers, had an in-depth analysis of cloud storage service technology development status and the specific situation of the cloud storage products abroad. At the same time, it analyzed the characteristics and disadvantage of the traditional storage technology, and puts forward a deployment scheme of distributed object cloud storage system which is suitable for enterprise.

Secondly,this paperput forward a feasible cloud storage service system frameworkbased onOpenstack, designed and realized a management system with the capability of providing basic storage management and application management. The content managementmodule is the basic of cloud storage service module, and it provides a cloud storage resource pool to control basic storage service functions, such as upload, download files, etc. In addition,the application management moduleprovides the functions such as daily maintenance, statistical analysis,cost settlement business.

Thirdly, this paper designed and realizeda third-party application interface to make the cloud storage system as a data store for WEB.WEB application use the third-party application interface in this paper to interact with the cloud storage system, as result to realize the multimedia files stored on the basis of the HTTP service

Finally, this paper has already done tests, within our expect, it not only has good storage service function, and also provide a function which is suitable for business operation and system operational. In addition, the development of third-party application interface can solve the problem of the volume increase of user request in the traditional small and medium enterprises, and validate the availability of cloud storage system in this paper.

Keywords: Distributed Storage; Object Storage; Openstack; Cloud Storage

目录

[摘 要 I](#_Toc374450219)

[Abstract II](#_Toc374450220)

[第一章 绪论 1](#_Toc374450221)

[1.1研究背景 1](#_Toc374450222)

[1.1.1云计算的概述 1](#_Toc374450223)

[1.1.2云计算概念的延伸--云存储起源 2](#_Toc374450224)

[1.2存储的基础知识与研究现状 2](#_Toc374450225)

[1.2.1传统的存储技术及其特点分析 3](#_Toc374450226)

[1.2.2国内外云存储技术发展状况 6](#_Toc374450227)

[1.2.3 SINA云存储标准 8](#_Toc374450228)

[1.3论文研究内容及主要结构 9](#_Toc374450229)

[1.3.1 论文研究主要内容与意义 9](#_Toc374450230)

[1.3.2论文章节结构 10](#_Toc374450231)

[第二章 传统网络存储技术及其结构比较 12](#_Toc374450232)

[2.1 分布式存储技术 12](#_Toc374450233)

[2.1.1 网络存储技术 13](#_Toc374450234)

[2.1.2分布式文件系统 18](#_Toc374450235)

[2.2 三种网络存储技术的对比与分析 21](#_Toc374450236)

[2.3 本章小结 22](#_Toc374450237)

[第三章 面向对象的存储技术概述 23](#_Toc374450238)

[3.1对象（Object） 23](#_Toc374450239)

[3.2 OSD (Oject-based Storage Device) 的基本组件 23](#_Toc374450240)

[3.2.1元数据服务器 23](#_Toc374450241)

[3.2.2 对象存储设备 24](#_Toc374450242)

[3.2.3 对象存储客户端 25](#_Toc374450243)

[3.3 对象存储文件系统 26](#_Toc374450244)

[3.4 对象存储网络结构 27](#_Toc374450245)

[3.5对象存储的特点 27](#_Toc374450246)

[3.6 本章小结 28](#_Toc374450247)

[第四章 面向对象云存储平台设计及第三方应用开发 29](#_Toc374450248)

[4.1 面向对象云存储系统的需求分析 29](#_Toc374450249)

[4.2 系统的整体框架设计 30](#_Toc374450250)

[4.3 云存储系统功能模块设计 32](#_Toc374450251)

[4.3.1 内容管理存储功能模块设计 32](#_Toc374450252)

[4.3.2应用管理模块设计 39](#_Toc374450253)

[4.4 第三方接口模块开发与应用 43](#_Toc374450254)

[4.4.1 应用需求分析 43](#_Toc374450255)

[4.4.2网络存储应用架构 44](#_Toc374450256)

[4.4.3权限认证（Authentication） 46](#_Toc374450257)

[4.4.4获取账号元数据 47](#_Toc374450258)

[4.4.5更新账号元数据 48](#_Toc374450259)

[4.4.6查询容器列表 49](#_Toc374450260)

[4.5本章小结 52](#_Toc374450261)

[第五章 云存储平台及第三方应用测试 53](#_Toc374450262)

[5.1 云存储系统部署环境说明 53](#_Toc374450263)

[5.1 内容管理存储模块测试 54](#_Toc374450264)

[5.1.1 租户注册功能测试 54](#_Toc374450265)

[5.1.2 基本存储功能测试 54](#_Toc374450266)

[5.1.3 查看账户明细 56](#_Toc374450267)

[5.2 应用管理功能模块测试 56](#_Toc374450268)

[5.2.1 查看系统租户情况功能测试 57](#_Toc374450269)

[5.2.2 查看租户消费情况测试 57](#_Toc374450270)

[5.2.3 管理租户权限测试 58](#_Toc374450271)

[5.2.4 运维管理测试 59](#_Toc374450272)

[5.3 第三方应用测试 60](#_Toc374450273)

[5.3.1 第三方应用测试说明 60](#_Toc374450274)

[5.3.2 WEB端网页上传功能测试 60](#_Toc374450275)

[5.3.3云存储端功能测试 62](#_Toc374450276)

[5.3.4 WEB端删除文件功能测试 62](#_Toc374450277)

[5.4 本章小结 63](#_Toc374450278)

[总结与展望 65](#_Toc374450279)

[参考文献 67](#_Toc374450280)

[攻读硕士学位期间取得的研究成果 69](#_Toc374450281)

[致谢 70](#_Toc374450282)

# 第一章 绪论

## 1.1研究背景

### 1.1.1云计算的概述

互联网发展的最近十年中，学术界与信息技术领域在以高效利用计算和存储的物理资源实现大范围的协同合作、资源共享来达到高效率低成本的计算为目标，前后提出的网格计算、分布式计算和并行计算，相继的结合虚拟化、效用计算、基础设施即服务(Infrastructure as a Service ，IaaS)、软件即服务(Software as a Service ，SaaS)的概念，云计算的概念正式被提出并得到产业界和学术界的广泛关注，同时也是现今甚至是未来几年最受关注IT技术之一和研究热点。

就如大家所知，云计算的兴起与发展带来了信息科技界的下一次革命；革命推动事物发生根本变革，引起事物从旧质变为新质的飞跃，云计算也使得人们在如何高效的利用计算和存储资源方面的思考和行为方式在性质和广度两方面发生了重大变化。那什么是云计算呢？在这里我引用维基百科的一种说法：“云计算是以互联网为传播媒介，按照按需、易拓展的服务交付形式来全面解决IT问题的计算方式，这种计算方式可以提供给计算机、服务器或者其它设备进行信息和软硬件资源共享，在云计算集群中各种应用程序共享这些计算资源，就好像是在相同的一个系统上运行一样。”

云计算概念的提出，对于产业界以及中小型研究机构也是一场至关重要的变革，解决了几个至关重要的问题；比如云计算可以减少与提供 IT 服务相关的成本，并且运营成本低，企业发展各项业务不再是仅仅依赖企业自己本身数据中心的物理设备，降低购置物理设备例如服务器、物理机等成本开销，这样，企业在整个运营过程不需要考虑在物理成本和维护方面的开销，企业的研发以及运营维护人员可以将工作重心放在企业价值创造方面。另外，高效灵活的云计算框架提供给企业和开发者更加简单的拓展自身的IT部署结构，使其更加适应瞬息万变的市场要求，让企业可以更加快速的占领市场，保证领先地位。

### 1.1.2云计算概念的延伸--云存储起源

2007年以来，互联网技术以及互联网产生的海量数据的爆炸性增长，硬件(磁盘技术)、存储方式、计算机资源和计算方式、人力维护成本的逐渐提高等等，已经成为互联网高速发展的新时代种种问题，从而提出了大数据时代下的各种新的需求。根据2011年来自IDC的一份科学研究报告指出，2013年全球范围内产生的数据总量已经达到2300EB，是2007年全球的数据总量的十倍，有权威的专家预测到2015年，全球产生的数据总量将会达到7000EB以上，与此同时，各大企业为了有力的支持其背后的商业行为，对企业所拥有的数据进行收集、分析以及处理，建立一个高效的计算、存储以及处理的平台已经变的格外重要，结合云计算概念的提出，如何安全快速的解决互联网产生的的海量数据信息，是我们亟需解决的难题，也是云计算向下发展需要解决的巨大挑战。

云存储不单单是字面上大多数人理解的“存储”的含义，它是在云计算概念上衍生发展出来的，分布式计算(Distributed Computing)、并行处理(Parallel Computing)和网格计算(Grid Computing)的发展从而衍生出云计算的概念，通过云计算技术，通过网络将大量的数据或者是计算机程序自动的分成无数的较小子程序交给服务器组成的大型云计算系统，进行分析处理后交给用户，因此网络服务提供商充当一个超级计算机的角色，在数秒之内处理了数以亿计的数据信息。云计算解决的难题之一是如何高效利用计算的空闲资源，相同的在云计算的存储能力上，也提出怎样提高存储效率的问题，从而，一个崭新的概念--云存储孕育而生，当企业部署云计算平台是为了处理和分析大量的数据信息和程序时，云计算系统此时也需要配备大量的存储设备，所以这个云计算平台就成为了一个以数据存储和管理为核心服务的云存储平台。

## 1.2存储的基础知识与研究现状

第三次工业革命--信息技术革命，综观人类以信息技术为代表的信息技术发展史，曾经先后有过两次的飞速发展信息浪潮，第一次是以信息处理技术为中心，以处理器的发展为核心的计算机工业浪潮，尤其是PC计算机工业，它使得计算机得到了迅速普及和高速的发展与应用；第二次是以信息传输技术为中心，以网络的发展为核心动力的互联网产业的大浪潮，互联网由一些使用公用语言互相通信的计算机连接而成的具有快捷性、普及性网络，比以往的任何一种通讯媒体都要快的大众传媒使得人们无论在何时何地都可以方便快捷地获取和传递信息。前面两次的信息技术浪潮的推动下，极大的加速了数字信息化的进程和发展，在网络信息爆炸式发展的时代，越来越多的信息活动转化为数字化信息，并且网格计算、分布式计算和云计算概念的提出，从而相继的引发了信息技术的第三次产业发展浪潮--信息存储技术浪潮。

### 1.2.1传统的存储技术及其特点分析

经历多年的科研和发展，在存储领域一代又一代的科研工作者进行了丰富的理论研究和科学实践活动，同时也验证了大量可行的方案，本节主要介绍存储技术发展过程中和本文有关的技术和相关知识。

1、集群系统存储技术

集群的概念相信在现代信息技术领域已经不陌生，在集群中，一组互相独立的节点或主机可以通过高速的互联网进行同步调的工作，就好像是处于单一的系统模式管理一样。集群是一种新型的技术，人们可以利用一组家庭中互相独立的计算机，通过高速的网络互连方式，就可以构建一个集群系统，系统的计算能力等性能可以媲美超级计算机的数据处理能力，利用较低的成本来获得高性能、可靠、灵活的计算处理能力是集群系统的好处之一。集群技术一直是信息科学技术的研究热点，特别是在云计算高速发展的时代，集群技术的好处更加明显。企业可以充分调动手中掌握的计算存储资源，利用集群技术将成本较低的软、硬件资源，通过虚拟化的方法来实现具有较高性能的数据存储和计算处理能力的计算机系统。对于客户端来说，集群系统的内部结构对用户是透明的，用户是不能直接看见集群系统的内部的具体结构，客户需要关心的只是集群系统带来的计算处理能力，通过统一的管理集群系统中的计算资源来达到高性能的计算处理能力。

对应的出现集群文件系统的概念，它是指架构在一个高效可扩充的服务器集群中的文件系统。在整个集群中的所有成员，不需要考虑文件是存储在单个成员的存储设备上，还是存储在集群成员连接的设备上，都可以通过访问协议来查看文件系统，用户可以通过系统的API接口进行数据存储。

集群系统的优势主要体现在一下几个方面：

(1)集群系统设计应用规模大，集群系统可以由数据处理需求适当地增减服务器(几十台文件服务器、文件服务器之间可以跨越几个路由器),灵活的设计集群架构来满足企业的业务需求。

(2)集群系统采用开放式的基础架构，通常使用分布式的操作系统来进行统一的协作与调度，集群中各个节点的地位都是对等的没有特殊的主次区别，功能也是完全相同，加强了文件服务器之间的协作，并且能够简单方便修改节点服务器的配置，拓展性高。

(3)集群系统釆用较好的容错机制增加了提高文件可靠性的手段，保证数据的高效可用。

数据信息量的增加和存储的集中处理给存储系统带来了巨大的挑战，这使得高效的存储不仅要有大容量、扩展性好、可用性高之外，还需具有可靠、安全方便管理等特点。因此集群技术得到重视，已经广泛应用于企业大数据中心服务器,为企业提供着易扩展、高可用性、易管理的存储服务。集群系统虽然有着许多优点,与传统的分布式文件系统相比,集群系统的可拓展性得到充分的体现，但是比如数据的一致性、动态扩展性等方面,还需要我们深入地去研究,更加优化其计算存储上的特性。

2、网格存储

Grid网格存储是目前的一个热门研究领域，正如这个项目刚开始的目的一样：将遍布全球的数以亿计的计算节点通过高速互联网Internet连接，并集成一个庞大的计算机系统，使其能够对用户透明、高效地完成复杂大型的计算任务。因此，在存储领域，研究人员按照网格在计算环境中把计算能力用于服务化的理念将存储资源也作为服务，提出了网格存储这一技术。

网格存储就是将所有的硬件存储资源虚拟化为一个存储资源池,这个共享的存储资源池就是我们说的网格。虚拟化和统一性管理的问题是网格存储的关键性的两个主要问题，存储网格可以对企业的数据进行统一的分析、查找以及管理，胜于单一的对当前物理资源的虚拟化实现，还可以更加高效的优化企业的数据远程管理操作和访问的性能，存储网格在近几年Internet爆炸性发展，海量数据呈现几何增加的环境下得到有效的利用，人们提出了面向广域网的数据管理方案DATA GRID方案，主要的策略是：在现有的数据库以及文件系统的统一管理的基础下，提供中间件，利用这一层中间件使得这些分布的、异构的系统能够进行统一高效的管理，使其形成一个统一的整体，共同对外对用户进行数据服务。

网格实际上是一种格状网，不需要单一的中央交换机来控制路由，从而使得我们可以高效的利用存储资源而不需要通过一个大型的中央交换系统，消除了交换系统的开销和瓶颈，因此网格在大小和行为上几乎不会受交换机需求的影响。网格的概念应用到计算机网络让我们通过动态分配物理资源获得高效的资源利用率。这些资源分布在多台分布式计算机包括存储空间，带宽等等。网格存储系统往往包括以下几方面：

(1)存储阵列：存储系统连接在ATA磁盘阵列组成的面向块存储的网关和服务器交叉存储网络上。

(2)通用虚拟层：对用户存储需要组成一个可用资源的逻辑存储资源池。

(3)数据冗余和可用性：系统在网格的交叉节点存储了数据的副本，这样冗余数据可以保证由于故障引起的数据丢失并进行恢复。

网格存储的几大优点如下：

①可靠性方面：网格提供多个路径给每个存储节点，这样的设计比较易于提供服务以及有利于系统出错后的数据恢复，给企业和用户提供可靠安全的服务。

②一致性性能方面：网格系统不仅提供高可靠性，由于它不需要大型的中央交换机，并且对多个存储路径应用负载平衡技术，为整个网络提供始终如一的高性能。

③扩展性方面：在实际应用中可以使用廉价的存储物理设备灵活的拓展存储资源池。

3、对等(P2P)存储技术

多样化的应用程序，传统的存储系统像客户机/服务器模式已经很难满足客户

的需求，从此新一代的高性能的存储系统的呼声也越来越高，相应的P2P存储系统就孕育而生。对等网络技术(P2P)，又称点对点技术，网络中的计算节点的地位是对等的，没有中心服务器的概念，通过高速互联网中计算节点的带宽和计算能力来实现资源的共享，网络中的每一个客户端既是一个节点，同时又具备服务器的功能。

对等(P2P)结构从体系结构方面来看，没有中心结构，全部结点之间地位对等

相互合作来完成任务达到用户的需求；从用户的使用方式来看，P2P系统中每一用户不仅向其他用户提供存储资源,也从其他用户那里获得存储资源；从上面可以看出P2P相对传统的客户机/服务器存储系统体现出以下优势：

1. 良好的扩展性；P2P网络中计算节点的客户端同时具备服务器的功能，他

们互相通过Internet连接来共同完成对等网络中的计算存储功能，同时这种网络结构不依赖中心结点,不会形成瓶颈，不易受攻击，可扩展性良好。

1. 适合大规模的存储；P2P的结构容易在广域网中组建大规模的分布式存储

系统，将PC或者是服务器对等的组织起来为用户提供大容量的数据存储业务，网络中的资源存储规模是多个节点的加权之和，通过各个对等的节点上资源来实现存储资源的贡献而摆脱了服务器硬件的瓶颈。

1. 良好的访问性能；在用户进行远程访问过程中各节点服务器并行处理用户

请求，均衡系统的负载以及根据系统的性能，用户可以根据情况访问距离最近的共享文件；

1. 廉价的成本；由于是硬件资源的高效整合和共享，P2P存储系统可以节省

大量的硬件成本和维护成本，实现高效的资源利用，同时容错能力强，可靠性高。

### 1.2.2国内外云存储技术发展状况

平板电脑、手机和多台PC设备的快速普及，使得人们意识到云同步和云存储服务在日常生活中变得日益重要。现今在云服务领域，苹果、谷歌和微软等著名的国外云服务提供商有着各自不同的工具满足企业和个人的远程数据备份、信息共享等需求，国外较早产品有Dropbox、微软的Windows Live SkyDrive、谷歌的Drive、iCloud等，国内应用较多的云存储产品是阿里云、金山快盘、百度云盘等。

1、微软SkyDrive

由微软公司推出的一项云存储服务，用户可以通过自己的Windows Live账户进行登录，上传自己的照片、word、excel、powerpoint等文档到 SkyDrive中进行存储，使用 SkyDrive 和免费的 Office Web Apps 在浏览器中获取免费 Word、Excel、PowerPoint 和 OneNote。不同位置的多个人可以同时处理同一文档。而且仅共享你所选的文档，与所选的人共享文档，并将其余文档设置为保持私有状态。

2、Dropbox

DropBox是由Dropbox公司支持的在线存储服务的网络文件同步工具，通过云计算在Internet实现文件的同步用户可以远程的共享自己或他人的文件和文件夹。不像微软和谷歌的移动平台应用程序，不能互相支持对方的移动平台，Dropbox是唯一一个支持Linux和BlackBerry智能手机的平台。Dropbox的Web应用程序的内置文件浏览器包括Office的常用套件、PDF、JPG等文件格式的图像、视频文件等。

3、苹果iCloud

iCloud是苹果公司所提供的云端服务，苹果设备上的系统备份、传输服务、音乐这些信息会自动推送到你手中的iPhone、iPod Touch、iPad，甚至是Mac电脑，iCloud将PC、移动平板、手机等产品线等元素有机的结合起来，联系紧密，这些设备可以用来存储图片、音乐文件、电子书籍、游戏配置文件等保存在苹果的服务器上，进行无缝共享文件，并通过无线网络推送到你的苹果设备上。

4、Google Drive

Google Drive是美国谷歌公司于2012年推出的一项云存储服务，它采用google的云存储基础架构，用户可以创建、共享、协同合作各类型的文件，对于协同合作同一任务的用户可以协同办公，它拥有强大的关键字、文件类型等搜索功能，同时还整合了Gmail、Google+等多项服务。

5、金山快盘

金山快盘是国内首款并行同步网盘，它是由金山软件基于云存储推出具备文件同步、备份和共享的存储服务，现在已经发展成为超过4500万的用户，并且覆盖了Windows/Mac/Android/iPhone/iPad/Web平台实现互通互联。

6、新浪云计算SAE

新浪作为国内发展较快的公有云研发的领头企业，从开发开始就借鉴了Amazon、IBM、Google等国外公司的公有云计算的成功经验，推出了具有自己特色的云计算平台，SAE是新浪推出的一款以Web开发语言PHP作为首选的支持语言，在Linux/Mac/Windows上通过Web对个人的代码进行在线的调试、编辑、部署以及开发，同时团队可以进行协同合作共同开发同一个项目，根据不同的项目权限，SAE提供一系列分布式存储服务供研发人员共同使用，包括分布式数据库集群、分布式文件存储、缓存服务等，大大的降低了开发的成本和运营成本，是一个典型的高性能安全解决方案。SAE在服务的可扩展性上，用户是以服务使用者的身份享受服务，资源上实现动态的可扩展性的思路。

### 1.2.3 SINA云存储标准

CDMI云存储标准是全球网络存储行业协会（SNIA）在2010年发布的第一个云存储标准，是云存储标准化工作的一个重要开端。现今已经编写并改进成为V1.0.2版本，据SNIA官网定义的说：“CDMI(云数据管理接口)提供给那些使用云计算并进行云中计算、存储访问接入的一个接口。”

客户端管理平台和数据中心服务器集群是CDMI的两大块集中讲述重点，这两个平台之间又通过描述一个数据交换的标准界面来联系。

图1-1 HTTP协议封装的REST接口访问

这个标准界面如图1-1所示通常采用web service(web服务)的HTTP协议来封装REST通信命令，首先客户端进行主动请求web服务，数据中心服务器相应客户端请求并对其提供web服务。这标准接口使得两种平台实现程序间的API对接实现交互，也就是客户端(平台上运行的应用程序)指令服务器端跑的应用程序，对云中存储的数据对象进行读取、创建、复制、更新和删除等若干操作(也就是我们熟知的Create，Read，Update，Delete即CRUD操作)，具体的过程如图1-2所示。



图1-2 CDMI接口访问存储数据

云存储标准正如道里云CEO毛文波先生所说“现今时代当一切成为数据的时候，云的做法就是将一切转化为服务，云存储标准是发展的，逐渐完善的。”目前的CDMI正在处于一个发展的阶段，云存储标准的制定是一个全球范围的参与、合作、创新及不断修进的长期过程，标准的制定与落地还有一段艰难的路要走，一些大的厂商如亚马逊、google、百度他们也有自己的一套标准，所以一套适合的标准还需检验其合理性与可用性，在接下来的云存储技术的高速发展过程中，云存储的标准还会不断的进行修订，标准的制定也是云存储发展的一个必不可缺的步骤。

## 1.3论文研究内容及主要结构

### 1.3.1 论文研究主要内容与意义

云存储是一项新兴的IT研究热点，近几年来国外已经出现许多典型的运用云服务降低运营成本和商业转型的企业案例，国外如亚马逊通过弹性云计算服务实现业务增长、Dropbox、iCloud等云存储产品已经拥有成千上百万用户。在国内，数码DV、平板电脑和智能手机等IT设备的快速普及，中小型企业海量业务数据处理的需要也对存储技术提出更高的需求；同时电信运营商企业内部有进行海量的客户数据存储、管理以及业务分析的需求，因此，综合国内云存储服务市场的内、外部需求，电信运营商利用自身丰富的带宽资源和庞大企业及公众用户数量资源，构建一个高效可用、可弹性拓展的云存储系统对外提供存储服务变得十分重要。

本文深入分析国外成功的云计算技术的案例，设计并实现一个提供基本存储能力的内容管理云存储服务平台；同时针对云存储产品的商业运营需求，以及国内云存储产品缺少的业务运营模块，设计了适合服务提供商进行业务管理的应用管理平台；最后设计并开发第三方应用接口，将本文的云存储系统做为WEB服务器的后台数据存储平台，大大较小了传统中小型企业WEB应用部署中WEB服务器的负荷，降低企业的运营成本，同时也验证了本文云存储系统的可行性与高可用性。

本文通过调研、需求分析、系统框架设计、系统功能模块设计、功能测试、应用测试等方法进行课题研究，主要研究的内容有：

1、详细分析传统的分布式存储技术的基本架构和关键技术及其特点，主要介绍了集群存储技术、网格存储技术、对等存储技术及其特点，对比传统存储技术的优势与不足，再分析新兴的对象存储技术的基本构成与特点，并深入研究典型的云存储系统模型；

2、提出了一个可用性高、弹性拓展的云存储系统结构框架，详细描述了云储存系统的功能模块，主要包括基本存储内容管理功能模块、应用管理功能模块。

3、针对本文的云存储系统架构图和功能模块，设计一个提供基本存储服务的内容管理云存储系统和一个提供给云存储服务管理员进行日常维护、统计分析、费用结算的应用管理系统。

4、设计并实现第三方应用开放接口，实现WEB应用服务器以HTTP方式与云存储系统进行交互，验证云存储系统的存储能力，并对系统进行功能测试。

### 1.3.2论文章节结构

第一章为绪论，主要介绍了云计算兴起的原因、云计算的概念和现时代环境下云计算国内外的研究状况，介绍了国外几种典型的云存储产品及现行传播较为广泛的云存储标准，最后描述了本文的主要研究内容以及意义。

第二章重点研究了传统的三种网络存储技术：直连存储、联网存储和存储区域网，分析了他们的典型的存储网络结构、存储机制原理和各自的技术特点，并介绍了分布式NFS文件系统和HDFS文件系统，通过对比三种传统的网络存储技术和新时代对存储技术需求，引出对象存储技术。

第三章对于海量数据产生新的需求，本章节提出了基于对象的分布式对象云存储解决方案，并深入分析对象存储的基本网络架构、组件功能及其特点，为云存储系统的设计和部署提供设计思路和基本方法。

第四章是本文云存储系统设计的主要部分，通过分析云存储系统的功能需求，提出系统的总体框架和功能模块，详细阐述云存储系统的基础存储功能模块和业务应用管理功能模块的设计思路和工作机制，设计用于统计分析的三类存储服务器的数据表格式和费用计算流程，最后设计了本文云存储系统的第三方应用接口并描述其对中小型企业的WEB应用环境部署的优势。

第五章是对本文云存储系统的基本云存储服务功能和业务应用管理功能测试，并用WordPress模拟第三方应用将云存储系统作为数据处理平台，进行第三方的云存储服务功能测试。

最后一章是对本文研究内容、设计过程的总结以及对本文设计的云存储系统下一步研究方向的思考。

# 第二章 传统网络存储技术及其结构比较

近十年来，互联网技术及其应用推动下的信息全球化，以及计算机应用技术的发展，DAS,NAS,SAN等网络存储技术通过将数据的存储从服务器转移到网络专用设备的方法,解决了以服务器为中心的传统存储方案所带来的问题；但是人们对数据存储的需求有着极大的改变，使得人们不得不重新评价以往的数据存储模型的性能，确保对以后发展的需要。不过，在此之前，我们先回顾一下存储系统模型经历的几个阶段，可以更好的理解本文设计的思路。

云存储是在传统的存储技术遇到新时代下的新需求的演变，是由传统的存储技术总结并拓展的一个新的概念，为了更好地理解选择基于对象的分布式对象云存储系统设计思想的优势，本章主要对传统网络存储技术(DAS,NAS,SAN)、分布式文件系统等分布式存储领域相关技术作概要性介绍，并对网络存储技术是否适应当今数据存储要求做了具体的分析。

## 2.1 分布式存储技术

分布式存储技术是区别于集中式存储技术的一项热门研究，它并不是单单的把数据存储在某个或多个特定的节点上，而是将企业现有服务器上的磁盘空间等分散的存储资源虚拟化为一个存储资源池，通过网络数据分散的存储在企业的各个地方。现今，分布式存储技术的主要关注的研究方向如下介绍：

1.可靠性方面

高可靠性作为分布式存储系统第一需要考虑的主要方面，是大多数系统能够提供始终如一的存储服务给用户的重点考虑的问题。用户将文件上传到分布式存储系统最基本的是拥有可靠安全的环境，因此保证数据的高可靠性是系统设计的基本目标。同时，系统还需要有一定的容错措施及机制，办证及时的处理一些如节点离线或失效、服务器故障、网络断开等引起的数据保存故障，当用户访问系统中的文件时，不会因为网络故障或部分节点不在线而使得用户不能成功的访问数据。总之，分布式存储系统应该避免集中式的部件设计，每一个部件都需要考虑到可靠性问题。

2.性能方面

除了基本的可靠安全的环境保障之外，分布式存储系统用户都应该得到高性能的存储服务，除去硬件、网络带宽限制的因素，系统设计要尽量的克服网络环境下的动态可变性和网络拥塞及断开、节点退出等引起的不可预知因素造成的服务性能下降。远程存储总是存在着延迟时间过长的问题，并且为了提高数据的访问效率，减少时延，系统设计要考虑用户进行远程控制的用户体验和存储服务的性能体验。

3.可拓展性方面

节点规模和数据规模的增长是分布式存储系统的重要表现之一，企业业务和数据的拓展对系统进行修改是经常发生的事情，系统的存储容量和节点的规模随着用户的需求而增长，系统的存储容量和系统的总吞吐率能够随着系统规模的增大而同步增大，因此系统的设计必须采用易扩展的拓扑结构、数据存储、组织策略适应存储节点和大数据的动态变化。

4.透明性

这里定义的透明性是指让用户和应用程序感觉到在本地存储空间一样，时延、网络服务质量等控制在一个用户可接受的范围内。分布式存储系统的透明性方面具体体现在位置、复制、迁移等方面。

用户创建的文件是通过名字来标识的，看到的是全局名字空间，创建文件时，系统自动选择文件放置的位置，用户不需要知道具体文件的物理位置，这就是位置透明性；系统往往在不同的存储节点上都复制了某个文件的一个或者多个备份副本，这些拷贝的副本的产生、分布和访问都是自动的，这就是复制的透明性；当分布式存储系统部分服务器出现故障、离线或网络不可达时，或者系统需要升级改进，用户不仅不会感知系统的故障，同时还不会在数据迁移过程中，出现数据是不可访问的情况，这就是迁移以及故障的透明性。

### 2.1.1 网络存储技术

网络存储技术（Network Storage Technologies）是为了解决传统的以服务器为中心引起的数据存储缓慢、故障率高等问题的存储方案，是基于数据存储从服务器转移到网络设备的一种解决方案。网络存储系统发展过程中主要有：直接附属存储系统、联网存储系统和存储区域网络。

#### 2.1.1.1 直接存储系统（DAS，direct attached storage）

直接存储系统(direct attached storage,DAS)从提出到现在已经有40年发展历史，作为最早被人们所认同的存储技术，DAS通用服务器经过网络连接存储器，存储器与服务器之间采用独立的外接式存储设备并通过标准接口技术如SCSI的I/O总线互连，将存储器的读写操作从应用服务器中分离，从而提高了总体的读取存储速度。如图2-1所示是DAS最早出现的网络存储结构，DAS存储结构中，客户端进行访问请求的响应步骤基本如下：

(1)客户机通过Internet发送访问请求到服务器主机；

(2)服务器端进行缓冲区查询；客户机需要访问的数据可以通过缓冲区直接通过适配器发送给客户机，或者请求翻译本地访问指令再发给存储设备。

(3)存储设备接收命令后拷贝数据到服务器缓冲区

(4)网络适配器的缓冲区从系统缓冲区拷贝数据后，发送给客户端。



图2-1 直连存储系统结构图

DAS完全以服务器为中心，DAS存储设备如磁盘、磁盘阵列直接挂载在服务器上，客户机每一次访问都要经过系统的文件服务器进行处理、分析并转发，服务器中的卷管理器对磁盘进行简单的管理，并作为一个逻辑存储设备呈现给应用程序和操作系统。DAS服务器在物理位置上可以十分分散，通过服务器进行连接十分困难，但是数据库应用、邮件服务等都必须直接连接到存储设备上，而通过SCSI连接的带宽限制，存储器与服务器之间的连接就会出现瓶颈。

存储资源的利用率低，当数据呈现急剧增长访问并发量过大的条件下，服务器端就不能够承受大规模的数据量增长，服务器的CPU负载、内存、I/O总线等负载过大，空闲的存储容量不能被其它主机共享，多个主机不能并行处理负载，从而产生系统的瓶颈。随着大数据时代的发展，数据备份、恢复对服务器的依赖将大大减小，相反系统的各个部分组件的耦合性方面却日显重要，在高可靠性、高拓展性、高安全性能要求的数据存储时代，DAS网络存储结构已经越来越不适应存储的需求，因此NAS存储结构被提出并被迅速的广泛应用。

#### 2.1.1.2 联网存储系统（NAS，Network Attached Storage）

联网存储系统NAS最初是由NETWORK appliance公司提出，就是为了解决前面DAS存储结构的“以服务器为中心”带来的许多限制，特别是解决了DAS的带宽等瓶颈带来的问题。它不同传统的存储方案以服务器为中心，NAS将网络中分散的、独立的数据整合为集中的大型数据管理中心，所以解决了服务器各种I/O、CPU、内存、带宽的瓶颈问题。

联网存储系统NAS是以“数据为中心”的一种数据存储模式，如下图所示，不同于DAS，NAS存储系统不用通过I/O接口总线连接某一个特定的服务器主机和客户机，而是直接通过网络接口直接连接，是独立于网络的一个专用文件服务器。通用的说就是客户机和NAS存储设备直接可以不通过文件服务器直接进行数据访问而不需要得到干预，从此避免了在传统“以服务器“为中心的存储系统中数据通过文件服务器转发处理过程中带来的延迟和阻塞，具有更快的响应速度和更高的数据带宽，降低服务器的性能要求，提高系统的可拓展性。

NAS是面向用户设计的用于数据存储的操作存储系统，系统可以支持NFS、CIFS、FTP、HTTP等多种网络连接和数据共享协议，并且设计了方便维护管理的浏览器管理界面。同时，鉴于企业的需求，系统管理员可以动态的增加NAS服务器的数量，NAS是一个易于安装、使用和管理，可拓展性能高的网络存储解决方案，NAS可以灵活的应用在任何的网络环境中，与比传统的DAS简单的将RAID存储磁盘阵列安装在共用服务器上，NAS存储结构示意图如图2-2所示，具有以下优势：

第一，NAS存储系统转为存储而设计的结构简化了通用服务器多余的计算功能，从而降低了硬件设备成本开销，同时由于NAS需要适合处理网络I/O接口的请求，专门设计强化了多线程处理能力、多任务网络操作能力的系统内核提高网络的相应速度和数据传输能力。

第二，NAS存储系统具有良好的可拓展性能，存储设备通过网络连接灵活摆脱了地理位置的拘束。

第三，由于独立主服务器外的专用的软硬件服务器，不会占用网络主服务器的系统资源，当企业需要增加存储设备时，不需关闭主服务器的任何进程，可以直接通过交换机即插即用增加NAS存储设备。

最后，NAS存储设备不受地理位置的影响拥有良好的可拓展性，在任何地方都可以通过网络连接起来。



图2-2 NAS结构示意图

#### 2.1.1.3 存储区域网（SAN，Storage Area Network）

存储区域网(SAN，Storage Area Network)是较为昂贵的一种存储解决方案，也是一种较为复杂的选项，即使很多企业在SAN初始阶段投入大量的费用，但是凭着它提供其它解决方案所不能提供的能力，在一定的程度上也为企业节省了后期不必要的成本浪费。

一种典型的SAN网络系统如图2-3所示，SAN网络系统中的存储设备不同于DAS和NAS那样，在DAS中存储设备直接连接到服务器构成以服务器为中心的数据存储结构，NAS存储系统则是将存储设备放到应用网络上，而SAN将磁盘阵列、磁带等存储设备互相连接起来形成存储网络，即存储区域网络；服务器通过光纤通道(FC，fiber channel)技术通过高速通道访问存储区域并互相连接起来形成数据存储专用子网，SAN 的体系结构中服务器可以访问 SAN 中的任何一个存储设备。



图2-3 存储区域网络SAN

近年来的Fibre Channel技术的快速发展推动着SAN的广泛应用，FC作为高速骨干网的通用技术，FC具备高达10Gb/s的带宽和数千个节点的寻址空间，通过光纤的传输介质网络范围也是可以达到10几千米，因此通过FC技术提供高速的通道给服务器访问存储区域网络，SAN具有NAS的所有优点，支持多种平台的拓展，方便高效的数据管理、备份和管理方式，SAN的优势具体体现在一下几个方面：

(1)高可扩展性。SAN存储网络中服务器可以访问系统中的任何一个存储设备，企业可以根据自己的业务和公司发展状况灵活的增加磁盘阵列、服务器等存储设备来增加SAN系统的存储资源空间、系统规模和提高系统的处理能力，同时在部署这些新增加的服务器和存储设备时物理地域的选择比较灵活确保了企业的运营成本。

(2)容灾恢复能力强。SAN网络是通过光纤通道进行连接，高速的数据访问可是实现多个服务器和多个存储设备的环境中两两之间的连接，并实现远程数据备份和容灾恢复。光纤通道的高吞吐率和远距离的传输特性，通过点到点(Links)、仲裁环(FC-AL)等多种网络拓扑结构可以创建多条路径任意进行数据访问，构成独立的数据存储集中管理，基于完善的存储管理机制动态的增加删除设备，将存储网络设备动态的形成资源池集中管理，形成高可用的容灾环境。

(3)备份能力强。光纤通道将磁盘阵列、存储设备以及服务器连接起来，存储数据流不经过LAN，从而摆脱了用户网络流量以及带宽的限制，提供多个数据流同时备份于多个存储设备的优势。

(4)高可用性。许多情况下，多数厂商都会出现这样一种情况，在一台服务器上出现的故障使得企业的数据丢失或者访问失败，SAN设计服务器集群避免单台服务器故障的数据失效问题，在SAN单台服务器失效时，FC高速通道实现多台服务器的对等存储，在故障的服务器上的数据可以通过服务器和存储设备之间的数据路径查找到其他服务器上的有效数据，确保数据的高可用性。这种数据路径是通过两个HBA(光纤通道主机总线适配器host bus adaptor)形成两个分主次的光纤通道，当一个光纤通道出现故障时，另一个通道通过管理软件可以自动的进行工作替换，正因为这种随时候命状态的备用数据路径和服务器群集软件保证了系统不会因为单台服务器故障导致系统的崩溃。

虽然SAN集合了DAS和NAS存储方案的几大优点，比如在SAN网络存储解决方案中你可以拥有一个完全冗余的具有高可拓展性的存储网络，翻译过来就是你不仅仅得到了只能在 DAS拥有的块级数据访问功能，还可以得到只有在NAS存储方案中几百甚至是上千T的存储空间，所以说SAN集合了DAS和NAS存储方案的最主要的存储优势。但是他却是有昂贵的成本代价和网络设计的复杂度两大主要不足，光纤信道合理的成本大约是 1TB 或者 2TB 大概需要五万到六万美金，硬件上昂贵的成本使得基于FC的SAN存储网络方案成为企业考虑的一大因素。

### 2.1.2分布式文件系统

分布式文件系统是一种服务器-客户端模式的文件系统，通过网络互连不同的节点，并在节点上部署管理模块和其所存储的数据。它可以同时服务于多个客户端，同一个全局命名空间可以被多个存储服务器使用。该文件系统可以实现智能分散数据和I/O。他们的基础结构如图2-4所示：

图2-4 分布式文件系统基础结构

分布式文件系统主要有三个组成部分：元数据服务器，数据块服务和客户端。这三个部分之间的互连是通过网络构成一个整体。其中，逻辑层服务器就是我们所谓的元数据服务器，它的作用是维护一个全局的命名空间。一般，会有1-2台服务器为整个集群提供逻辑层服务。数据块服务器对应的是存储层服务，它是分布式文件系统的核心，是用来存储数据的地方。在一个集群中，该服务器规模非常庞大。它们把数据块以及一些校验数据按照一定的规则以文件的形式存在存储介质上。客户端提供接入层服务。它是用户和文件系统的桥梁，是用户能够透明的进行I/O操作的基础。

#### 2.1.2.1 NFS文件系统

NFS文件系统是一种可以使得处于不同操作平台上的用户共享一个文件系统。这样用户可以方便地使用跨平台系统下的文件，使得资源方便共享，极大地扩展了使用空间。NFS具有的典型的分布式文件系统特性：客户端/服务器架构，每个节点之间通过RPC通讯，并且都可以作为客户端或服务器。NFS文件系统的核心机制是采用RPC机制实现分布式网络服务。其中主要包括两个方面：挂接和访问NFS文件系统，客户端、服务器端进程管理。

NFS文件服务器是具有较强容错能力的优点。并且服务器崩溃，状态信息也不会丢失。这是因为NFS服务器是无状态的，不保存内存中的缓冲状态。但是同样也导致当模块缓存失效时，NFS服务器无法识别该缓存数据的拥有者。一般的NFS系统不具有复制功能，一旦NFS服务器失效，系统将停止为客户服务。为此IBM提供了一个完善的解决方案：HA-NFS。其结构如图2-5所示：

图2-5 HA-NFS结构

两个HA-NFS节点之间相互独立，提供无交集的服务。但是两个节点之间又相互备份。其中，一个HA-NFS节点包含两台NFS服务器，分别是主服务器和辅服务器，两个共享一个总线。正常情况下，通过总线，只有主服务器在工作。当主服务器失效后，辅助服务器会立马替换主服务器开始工作。

#### 2.1.2.2 HDFS文件系统

HDFS的核心节点是namenode，对应分布式文件系统结构的元数据服务器。它主要有两个作用：一方面，管理HDFS文件系统的命名空间，另一方面，维护本系统的datanode和数据块映射。但是两者在所要承担的开销不在同一个数量级上。在namenode本地磁盘上，会永久保存命名空间所需要的数据信息，如，目录树及其目录的元数据。但是数据节点和数据块的映射不会被永久保存。这些信息只是在系统启动时，通过各节点的通信建立起来的。通过这种设计方式带来的便利是：一个内存仅仅4G的namenode服务器，就可以维护海量的文件盒目录。

如图2-6所示，Datanode对应分布式文件系统基本结构的数据块服务器，它用来处理数据传输。通过周期性的与namenode通信，以获取本数据节点上的数据块信息。一个datanode会存储数据块ID、其内容以及相互的映射关系。并支持用户的查询请求。为了防止数据丢失，将同一个数据块在多个datanode上备份；同时一个datanode节点上至多有一个数据块的备份，避免了空间的浪费。

图2-6 HDFS数据通信流程图

HDFS的namenode是整个系统的关键节点，它是将datanode上的数据块恢复成文件决定性部分，它的失效将对整个系统带来毁灭性的灾难。因此，HDFS提供了两种方法来保证namenode的安全性和有效性。

第一种，直接备份namenode：运行一个后备服务器namenode-slave。通常namenode-slave与namenode不在同一个物理服务器上。文件系统的空间镜像和文件操作的日志文件会周期性被它合并保存。这些被保存的数据在必要的时候可以直接引导文件系统工作。

第二种，首先，尽可能地长期保存namenode上文件系统的元数据，然后就是把namenode的持久化信息写到多个文件系统备份。

HDFS的datanode节点会在物理物理磁盘上分块。块的默认大小是64MB。这样做的优点是显而易见的，分布式系统只需和块进行交互，极大地提高了管理的便利性。另外，根据副本机制，仅仅需要对块备份和容灾，提供了系统的稳定性和灵活性。

## 2.2 三种网络存储技术的对比与分析

在2.1.1节中详细介绍了DAS、NFS、SAN三种传统的网络存储技术，三种存储技术的几个维度的对比如下表2-1所示，开放式的直接存储DAS系统的不足主要表现两个方面；首先，I/O总线方面，DAS存储系统是通过SCSI总线连接存储设备和服务器，而SCSI总线的限制只能连接最多16个设备，不适应互联网中大数据的生成速度；其次，DAS存储系统主要是服务器与存储设备物理上的叠加，用户访问的海量数据都要经过服务器和存储设备间转发多次，数据的处理将消耗大量的CPU资源而降低服务器的相应速度，从而使得存储设备拓展和共享都比较困难，服务器的限制成为系统的瓶颈。

表2-1 DAS、NAS及SAN存储技术对照表

|  |  |  |  |
| --- | --- | --- | --- |
| 项目 | DAS | SAN | NAS |
| 部署安装 | 容易 | 困难 | 直接插用即可 |
| 兼容方面 | 较好 | 较差，标准不一 | 可实现跨平台 |
| 拓展性能 | 弱 | 强拓展性，可以没有限制拓展 | 中等 |
| 成本 | 低 | 高部署成本 | 中等 |
| 网络复杂度 | 低 | 复杂 | 中等 |
| 维护成本 | 高 | 较高 | 低 |
| 容错能力 | 弱 | 强 | 中等 |

NAS联网存储系统中服务器和NAS存储设备之间是直接连接到网络上，首先服务器使用公用网络向NAS存储设备发送从客户端获取的数据请求，然后NAS存储设备使用网络将数据传回给服务器并完成客户端请求的相应操作；因此NAS存储技术虽然利用网络资源实现了数据的共享和高速存储，但是在请求数据量庞大的情况下，有限的网络带宽和负载将限制存储系统的存储效率和响应。同样的SAN存储区域网虽然集合了DAS和NAS存储系统的主要优点，但是在网络设计部署成本上，硬件上昂贵的成本使得SAN存储方案不适应大数据时代的进程，另外SAN网络设计的复杂度高，安全性能较差。

## 2.3 本章小结

云存储技术是在传统的存储技术遇到新的需求后演变并发展而来的，本章节是本文的基本理论基础，主要分析了传统的网络存储技术的基本结构及其特点，主要包括直接附属存储DAS、联网存储NAS和存储区域网络SAN；另外，还对比分析了三种存储方案的特点与不足，三种传统的存储方案还不能够满足数据量爆炸式增长时对数据处理能力的需求。

# 第三章 面向对象的存储技术概述

互联网及其应用带来的海量数据的爆炸式增长，对网络化存储的要求也越来越高。传统的网络存储结构已经不能满足数据存储的需求，然而基于对象的存储技术将存储资源连接起来形成linux集群系统，OBS系统中计算节点可以并行访问磁盘等存储设备，集群中的服务器节点能够解决大量并行的请求，并且系统中的元数据可以实现数据访问共享，避免了系统的瓶颈，提高系统的可拓展性。在此背景下，对象存储开始发展壮大。本章节主要研究对象存储的体系结构和基本组件，分析对象存储系统的网络结构及其特点。

## 3.1对象（Object）

搞清楚什么是对象，对弄清楚对象存储系统有着重要的意义。对象（Object）是对象存储的基本单元，这个基本单元与C++中“对象”的概念相似实际上包含了数据、属性及其方法。它的特点是可变长的，灵活性高。可根据应用的需要，存储数据库记录、文件、视频、图像等任意类型的数据。另外，每个对象维护自己的属性，从而大大降低了系统的负担。

## 3.2 OSD (Oject-based Storage Device) 的基本组件

基于对象存储设备OSD(Object-based Storage Device)，是在2004年由美国国家标准学会ANSI推出的1.0规范版本，OSD基于对象的存储设备给存储带来了新的革命，不同与以往的存储设备，对象存储同兼具SAN高速直接访问磁盘特点及NAS的分布式共享特点。对象存储结构组成部分（对象、对象存储设备、元数据服务器、对象存储系统的客户端）。下面对OSD各个组成部分进行简要介绍。

### 3.2.1元数据服务器

OSD内存在着MDS元数据服务器部分，MDS主要进行管理和存储OSD中的文件的元数据，在OSD存储的内部，元数据主要分为两种，分别是元数据目录和元数据文件两部分，元数据目录中存储的有目录（名称，存储目录的路径，ID号，访问该目录的控制权限等）内容。而元数据文件则是存储含有一些关于文件存储的一些内容，包括：文件的访问属性及文件的对象信息等等。

对于元数据的管理方面，OSD采取了MDS集中式统一管理元数据的方法，OSD将元数据集中在一起交给OSD统一管理，存储元数据，OSD先把存储进来的元数据转化成标准形式，这个标准形式就要符合服务器存储数据的标准格式，转化完全后将该数据路径名作为元数据的标志，运用哈希算法，将元数据存在同一个目录下集中式的存储系统。集中式元数据服务器结构如图3-1所示：

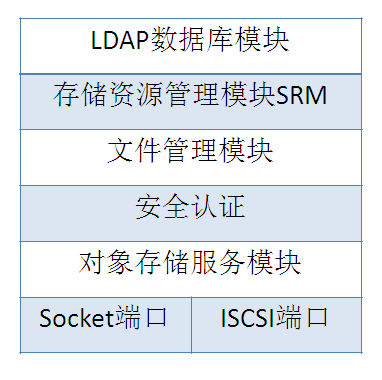


图3-1 集中式元数据服务器软件模块

各部分主要功能是：

1. LDAP数据库模块：实现了元数据目录服务。目录服务是一种特殊的名称服务。它具有迅速响应、支持大规模查询和同时提供复制功能的特点。通常是树状结构。
2. 文件管理模块：利用LDAP数据库，管理整个系统的命名空间。
3. 安全认证模块：负责认证各种条件。

元数据服务器也增加了数据传输的安全性，在网络中，存储系统中的每个用

户拥有不同的权限，要执行的命令也不尽相同，为了消除可能存在的操作混乱或者是命令间错乱或者是安全隐患，元数据服务器则是提供了一个安全防火墙，消除了这些发生的可能

### 3.2.2 对象存储设备

作为OSD对象存储设备，存储设备OBSD自然很重要。显而易见，存储设备就是磁盘，计算机，存储器等等。刚才讲的MDS会结合OSD和客户端返回的结果，根据返回值决定分布策略。

OBSD的软件模块结构，如图3-2所示：不同于元数据管理，OBSD直接向用户提供存储的接口，用户在访问存储设备时先向元数据管理器发出申请，在元数据服务器认证正确的情况下，OBSD则再次向用户提供身份认证，这也正是OSD安全性比较好的体现，每个服务器都要验证用户信息，极大的保护了用户的安全，认证结束后就执行用户的命令，对存储的内容进行读写，读写内容还包括对象的ID，偏移地址和长度。OBSD包括自己的CPU、网络接口等，是个独立的个体，主要功能有：

1、存储数据：主动存储数据或者结合应用程序存储相应数据。

2、智能分布：OSD系统具有强大的自适应功能，轻松的实现自我管理、自我优化存储格局和均衡容量等。并且OSD系统可以对对象数据进行预读取，加快了I/O相关操作速度。

3、元数据管理：直接在OSD上对存储系统中的块相关的元数据进行管理。大大降低了计算开销，减轻了网络和CPU等压力。

4、安全管理：OSD极大地解决了数据安全的问题，不同于以往的存储方法，OSD设立了单独的管理权限的系统，存储设备，计算机节点，计算机节点命令等都要经过存储系统对其进行认证，每次要访问用户对象都要进行认证，这样的安全水平给了用户最大程度上的保护。



图3-2 OBSD软件模块

### 3.2.3 对象存储客户端

客户端是可以向用户提供Linux文件系统的访问服务。用户在linux终端上发布命令，如cp，ls，rm，mkdir等对存储对象操作时，存储系统就开始启动存储数据。客户端结构主要有五个部分应用层、虚拟文件系统、客户端文件系统、数据存储模块以及接口模块。

如图3-3所示，应用层位于顶层，是应用程序访问的接口；虚拟文件系统VFS是Linux下的虚拟文件系统，通过屏蔽Linux下多种不同的文件系统，达到统一使用的效果；客户端文件系统CFS是Linux下自带的文件系统，兼容VFS，可以访问MDS的元数据；客户端对象存储、元数据服务器存储模块是用于处理、封装不同类型的命令；而接口模块处理本地文件与文件信息的显示，并是MDS和ODS通信的桥梁。

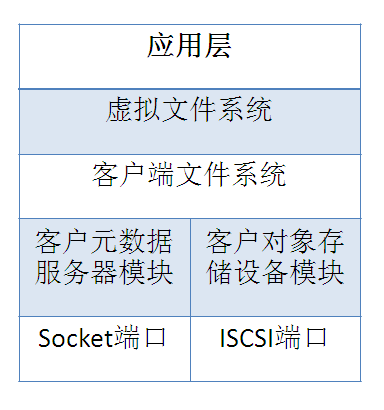


图3-3 客户端软件模块

## 3.3 对象存储文件系统

对象存储中文件系统能够接受用户发送的命令，能够与用户数据交换，并为用户服务，按要求提供数据，能够保持数据更新，能够同时满足多个用户使用。它主要用于连接访问节点，使得存储系统的操作系统能兼容POSIX，根据用户实现数据读写。对象存储文件系统主要功能如下：

(1) POSIX接口：文件系统兼容POSIX，使得客户端能够对服务器上数据进行打开、读取和写操作。需要支持对平台的应用，如Unuix、Linux。并能够对某一文件进行特定操作。

(2)引用缓存机制：文件系统必须有缓存机制，这保证各模块对常用数据处理的效率，减少传输等待时间。同时，缓存机制可以实现几个操作组同时协调工作，提高OSD上工作效率，比如，将MDS的元数据信息和权限验证结合，使得客户端可以更快的得到信息。

(3)数据条块化处理：文件系统能够将一个大的对象分解成若干个组件对象，并且让不同的模块同时处理。

(4) ISCSI协议：文件系统必须兼容ISCSI协议。这是传输OSD上数据的必要协议。

## 3.4 对象存储网络结构

网络结构的主要作用是使客户端到OSD和MDS之间可以通信，主要有三种网络协议：

1、RPC和LDAP协议：轻量级目录访问协议(LDAP，Light weigh Directory Access Protocol)，支持TCP/IP访问。远程过程调用(RPC，light weight Remote Procedure Call)的作用是服务于OBS（Oject-based Storage）系统客户端和MDS之间的通信。它的优势是使得客户端对元数据访问请求速率更快。

2、ISCSI协议：ISCSI协议是传输协议，规范传输内容到OSD的协议。通过ISCSI接口和SAS接口得以实现，将数据读写命令和数据一起传输，并且使用相同的方式传输。

3、常用的网络协议，比如，实现存储系统和节点间同步的网络时间协议（Network Time Protocol，NTP）；具有简化地址并可以分离客户端和存储系统分离的域名服务协议（Domain Name Server，DNS）。这些协议的运用，对OBS的性能有着极大促进的作用。

## 3.5对象存储的特点

OSD结构特性决定了对象存储对比传统存储结构的系统，具有传统存储结构以下几方面的优势：

1、方便管理：一方面，工作量的变化，OSD中对象的自我管理，分担了系统管理的工作量，减轻了负担。对于管理员来说，工作重心可以仅仅放在逻辑视图上，而忽视存储设备细节。另一方面，管理的灵活性强，无需额外的调整分区便可实现对OSD的存储系统容量。此外，OSD可以自动统计应用模式，并自适应的提高设备和访问效率和空间利用率。

2、可扩展性好：OSD数量的增加可以加大处理能力、I/O带宽。这种特性保证了系统的良好的可扩展性。

3、高性能：OSD解决了传统结构的存储分配功能。从而使性能获大幅提高。

1. 安全性好：由于对象作为基本存储单元，并是OSD独立管理。因此，可以很容易的设置安全访问策略，建立对象、对象组、OSD设备等层次访问机制。

5、数据共享更简单：OSD通过标准接口可以与任意客户端交互，而统一的命名空间又保证了数据的一致性，使得数据共享更方便。

OBS是存储技术的突破，对比以往的NAS和SAN技术，结合了NAS和SAN的优点，摒弃了它们的缺点。比如，NAS中的I/O路劲过长效率不高，SAN解决了NAS中的这个问题，用专用的元数据服务器处理数据，但是元数据服务器要管理所有的数据，当系统变大的时候，这就成了系统发展的大瓶颈。我们的OSD利用MDS和OBSD结合的办法解决了NAS中的问题，方便，高效的处理数据，综合了NAS和SAN的优点跨平台的数据共享，SAN的优点高拓展性。对象存储必然是未来发展的一个重要方向。

## 3.6 本章小结

本章引出了面向对象的存储技术，重点介绍了对象存储系统的重要组件，包括元数据服务器、对象存储设备以及对象存储客户端；深入研究对象存储系统的网络结构和特点，基于对象的存储技术优势适应数据爆发式增长的大数据时代对存储技术的需求，为本文系统设计提供了基础模型和设计思路。

# 第四章 面向对象云存储平台设计及第三方应用开发

电信运营商内部每天需要存储和处理的数据量(包括业务结算、计费等)增长都在1~2T，总数据存储容量也达到200~300T，因此全国数据量更加客观。无论对于企业内部的数据处理，还是对外提供服务的需求，因此构建一个高效可用的云存储服务平台显得尤其重要。

本章是云存储系统详细设计说明的主要章节，首先分析云存储系统的功能需求，其次提出一个可行的云存储系统技术框架和网络部署架构方案，再次阐述了系统功能模块设计的设计思路和具体流程，最后设计并开发第三方应用接口实现第三方应用与云存储系统的交互。

## 4.1 面向对象云存储系统的需求分析

根据本论文的项目合作对象--中国电信广东研究院的需求，立足中国电信集团在云计算的整体部署，本系统设计针对的对象群体主要是个人、家庭以及企业，为他们的个人或企业业务平台提供海量的数据存储能力和自助服务能力。

本文设计的云存储平台目标一方面是对用户提供高可靠的、可弹性拓展的便捷存储服务，另一方面对服务提供商提供有效的存储管理方式，简单的可操作界面和管理员管理门户。

因此，本文云存储服务管理系统的功能需求如下所述：

(1)内容管理：用户可以在应用自主门户上进行注册(创建用户申请)、修改、注销申请、创建容器、上传对象文件、下载、复制、删除文件等内容管理操作。

(2)应用管理：服务提供商的管理员进行运营管理的需求，运维人员通过系统管理界面实现云存储业务的运营工作，包括用户管理、统计分析、费用计算等。

应用管理主要提供应用注册申请的审批、冻结、恢复、统计和计费功能，及实现这些功能时应用与租户、资源信息对应关系的管理。

①应用注册申请的审批：系统管理查看应用注册申请，并可对申请执行审批操作。

②应用申请列表查看：具有待审批列表，已审批列表，支持按照关键字进行应用申请的查询。

③应用冻结：云存储系统可对应用所对应的租户进行冻结。

④应用恢复：系统管理员可对冻结的租户进行恢复。

(3)统计分析管理；作为电信运营商，中国电信面向中小企业用户和个人开发者提供安全、可靠、实时的数据存储这项服务，所以需要建立一套健全的运营模式；而对于设计者来说，需要提供一个界面风格友好的可视化界面，对用户所使用存储空间、请求指令次数、下行流量三项服务进行灵活、实时的计费，主要包括：

①提供所有用户访问的详单，记录信息包括但不限于：用户标识、访问时间、IP地址、操作内容、状态信息、异常报错信息等。

②提供单个用户使用（消费）详单统计功能。

## 4.2 系统的整体框架设计

根据上文分析的云存储系统需求，本文设计的云存储系统主要采用三级架构，如下图4-1所示，包括最上层的内外部一些应用外，主要是由云存储能力平台提供内容管理服务；云存储应用接入平台提供给管理员权限进行平台的资源管理、系统账户管理、日常维护、统计分析的管理能力。



图4-1 云存储系统技术框架

底层是云存储物理资源虚拟存储池，是由磁盘、服务器等存储设备虚拟化的存储资源池，有多个存储节点集群化管理，存放对象的物理设备组成。第二层定义为云存储能力平台，负责存储能力的提供，用户对存储空间资源、容器以及对象进行管理，提供对象存储、文件存储、元数据存储的分布式可拓展的基础存储能力，通过注册申请的用户可以通过自主服务门户进行相关的内容管理，创建容器、对象等基本存储服务操作，同时云存储能力平台提供包括租户、接入邀请码、容器管理以及租户、容器、对象的存储策略控制。上层是云存储应用接入平台主要提供应用管理平台的门户管理，站点管理、系统资源管理、日常运营维护、用户操作统计分析等。当用户在门户进行注册、修改、注销申请等自助服务操作，系统将为管理员提供管理入口登陆管理门户，即可以完成相应的平台管理操作。

本文设置的云存储系统网络结构图如下图4-2所示，来自互联网外部的用户请求验证经过认证服务器的权限认证，请求通过代理服务器(Proxy Server)进行转发给认证服务器，权限认证通过后，代理服务器中的account、container和object控制器被调用，而在存储节点磁盘设备上主要设计了账户(Account)服务器、容器(Container)服务器、对象(Object)服务器三类存储服务器，当控制器被调用，可以获得account的容器(Container)列表，进行对象(Object)的查找。本文实验系统由于资源有限，只是使用一台服务器作为代理服务器，在大规模的数据存储系统应用中，可以进行代理服务器集群，这是一个有效的负载均衡的改进方法。



图4-2 云存储系统网络结构图

结合本文上节分析电信业务的功能需求，本文系统的三层架构设计提供了以多租户为基础的多业务承载，可以实现多用户以及多业务环境下共用相同的系统或程序组件，租户之间共享物理资源节点，它们的数据都在逻辑隔离的存储空间，拥有独立的进程管理相应的服务，租户可以对容器进行创建、删除、恢复和属性管理等基本操作，还可以获取包括对象大小、个数以及容器中对象占用的空间等信息。

## 4.3 云存储系统功能模块设计

### 4.3.1 内容管理存储功能模块设计

本文云存储服务系统的重要功能之一是提供给企业或者开发者一个安全、可靠的在线存储功能，内容存储管理模块实现主要功能是提供给用户申请与注册、对象管理、容器管理、文件存储等基本的存储功能，在本小节中主要介绍本文云存储系统的基本内容管理存储功能的具体实现。

#### 4.3.1.1 模块功能

内容管理模块的基本功能是租户进行基础的存储功能，如账号申请、创建容器、上传对象、数据复制等基础存储功能，其基本流程如下所述：

1. 账号申请与注册

用户向服务提供商发出账号申请请求，获得租户权限后向验证服务器进行权

限验证，权限验证通过后经过代理服务器转发，代理服务器上的account、container和object控制器被激活，租户就可以登陆云存储系统进行基本存储操作。

1. 创建容器

租户发出创建容器请求到代理服务器，根据租户的权限并请求路由到存储资

源池，并在存储资源池找到合适的地方创建容器，最后返回容器创建结果给租户。

1. 对象上传

租户发出要在某个容器内写数据的请求，代理服务器查询到租户需要写数据

所在的容器并相应的转发请求，对象服务器接受到请求后接受租户的数据，并在存储资源池以对象的哈希值和操作的时间戳保存存储路径返回给代理服务器，租户在下一次写数据操作时可以通过代理服务器重新定位对象的位置。

1. 文件下载

租户发出读数据的请求，系统通过代理服务器进行检索和定位，如上文所说

代理服务器根据对象服务器返回的路径重定向到资源池，并把结果返回给租户，租户就可以进行文件下载操作。

1. 数据复制

在存储节点也是具备数据复制到另一逻辑存储资源池的功能，复制是把租户

的最新版本的数据拷贝成另一副本到云存储系统，当某台服务器上的故障导致用户读取数据失败或者是网络原因造成的事故时，办证了系统服务连续性。

#### 4.3.1.2 多租户架构

作为电信运营商，云存储作为一项服务提供给中小企业用户和个人使用。同时电信具有较大的基础设备和带宽的优势，利用自己自身资源优势构建一种可弹性扩展和灵活管理的云基础架构的同时，需要考虑到将存储资源以“租赁”的形式交给用户，而电信庞大的用户群体使得设计上出现“多租户”的方面的考虑，下面介绍一下多租户概念。

多租户模型

站在电信云服务提供商的角度看，对于具有多分布的云存储系统来说，如何将海量的数据分配给庞大的用户群并保障在不同用户之间的资源分配、共享以及隔离等问题，提高用户的存储服务满意度，是搭建本文云存储系统必须考虑的问题之一。

对很多人来说，多租户可以说是一个比较新颖的概念，可是这一概念很早就被提出。多租户一般来说指的是一个单独的实例可以为多个组织或者是用户服务，并且使得这多个组织用户的数据隔离，而这里多个组织服务就叫做多租户。

云计算成为热点的时代，多租户概念的出现使得人们考虑到如何让虚拟的共享资源池中的数据中心怎样在同一系统架构下提供不同的可定制化服务，并保障客户之间的数据的安全隔离。

在云存储系统中本文采用租户、子租户和用户3层资源分配体系实现多租户的资源分配，如图4-3所示，租户之间采用物理隔离的方法独占租用的物理设备比如服务器某个磁盘空间、磁盘组、网络资源等，其他租户则不能使用。子租户从属于一个租户，他们之间采用逻辑隔离的方法，即子租户与同一租户下的其他租户共用一个物理设备，云存储系统采用自动分配的技术为子租户分配逻辑存储资源池，标示以全局访问空间，子租户携带指定的全局访问空间标识访问虚拟化的存储设备，其存储的文件或对象可以通过存储系统的虚拟化感知能力在租户物理存储设备飘动。同样的用户从属于子租户，他们采用逻辑隔离的方法，通过访问权限控制等技术使用户间可以共享以及分享存储资源。在这种架构中每一个租户、子租户、用户之间采用独立的不同层级策略，提升相应的服务质量。



图4-3 多租户的资源分配模型

#### 4.3.1.3 存储服务器

存储节点(Storage Node)磁盘设备上主要设计了三类存储服务器，主要包括账户(Account)服务器、容器(Container)服务器、对象(Object)服务器，三类服务器的数据存储操作的请求与响应的具体过程如下图4-4所示。



图 4-4 数据请求与响应流程图

Account发出请求处理容器的信息例如(容器的位置查找、建立时间、删除时间等)，Account Server内部进程形成相应的存储路径并发送给Account。相同的，当扫描到Container的请求指令，通过Container Server查找对象的列表可以知道在某个容器中存的有哪些对象、对象的个数、容器的具体情况等信息，并且对象的信息是以SQL数据库的形式存储并备份。对象服务器的主要功能是存储、删除和搜索本地存储设备上的对象，在具体的物理磁盘device下的文件路径的目录结构包括账号、容器、对象的存储目录，在object目录中，object的数据(data)和元数据(meta)放在由对象的哈希路径名称name\_hash\_path构成的目录下，partition目录由suffix\_path的目录和一个hashes.pkl文件组成。

通过对中国电信的一些调研情况来看，选择一个适合的云存储解决方案主要有以下考虑：

首先，内部数据的增长迅速，电信运营商提供的往往是7\*24小时的服务，内部需要的存储和处理数据量非常庞大，各个省公司的结算、计费等数据量的增长都在1-2TB,总数据存储总量基本是在300-400TB；其次电信运营商目前存储数据的能力还只是一个月的水平，这样的存储水平在业务分析是远远不够的，不利于业务发展，成本的压力和集中化建设的管理等需求，开源的IaaS云计算平台使得电信在开发云存储应用上存在可能，目前适合建立一个面向公众、企业和开发者的弹性伸缩数据云存储平台的开源软件主要有Eucalyptus、ConVirt、AbiCloud、OpenNebula、Hadoop、Openstack、XenServer等，本文进行了一些调研作出选择，无论从功能上的虚拟化程度、存储能力、二次开发的难易程度、界面的友好程度等，还是结合电信商用运营性价方面，Hadoop和Openstack架构都是现今流行的云存储开源软件，环境配置简单、开源二次开发文档详细、具备强大的技术支持团队，本文对于两者的衡量具备以下几点考虑：

1、服务器单点故障方面。

Hadoop的分布式文件系统HDFS和OpenStack对象存储都有着相同的目的，实现快速、联网高效可拓展性的存储，但是HDFS采取了与GFS一样的主从结构，HDFS系统中有一个中央系统来维护文件元数据(在hadoop中称为节点Namenode)，因此拓展到一定规模的云存储环境容易出现单点故障。

2.多租户架构方面。

HDFS在设计的时候只是针对大文件进行优化存储处理，在多并发的情况下每次只能有一个文件进行写入，这种结构不适合多租户同时进行存储服务的大规模环境，相对的openstack的对象存储项目swift具有服务器集群的设计，数据使用zone进行物理位置上的隔离，是一个多租户的云计算基础架构，在多并发的环境下多个文件写入以最后一次操作为准，保障了存储服务质量。

综合上面所述多种因素诸如功能需求、应用对象需求、开源云计算基础架构特点等分析考虑，本文选择开源项目openstack的云计算基础架构为蓝本，本文将使用openstack的对象存储子项目swift进行参考设计。

#### 4.3.1.4 服务器数据库

本文使用的对象存储系统，通过Account和Container对数据进行逻辑上的区分，对用户的数据进行逻辑上的隔离，从而简化处理文件层次较多带来的困难，因此，设计上对比其他文件系统是没有文件夹嵌套的概念，采用以目录式的结构，其基本目录存储结构如图4-5所示，采用<Account\_name>/<Container\_name>/<Object\_name>的基本目录存储结构。

图4-5 存储服务器层级结构示意图

在账户(Account)服务器、容器(Container)服务器、对象(Object)服务器三类存储服务器中，本文使用轻量级的数据库SQLite建立了三类的数据库表，分别是Account、Container、Object三种类型的数据库。

在Account的db文件中设计包含了4张表如下表4-1、4-2、4-3所示分别是：记录Account名称、新建时间、Container个数等相关信息的account\_stat表；记录Container的名称、PUT和DELETE的时间戳、Object个数等信息的container表；记录Account中到来的同步数据选项和记录Account中推送的同步数据项的两个表(表incoming\_sync和表outcoming\_sync)。

表4-1 account\_stat的数据项结构类型

|  |  |  |
| --- | --- | --- |
| Key | 含义 | 数据类型 |
| created\_at | 创建时间 | TEXT |
| put\_timestamp | PUT请求指令的时间戳 | TEXT |
| delete\_timestamp | DELETE指令的时间戳 | TEXT |
| container\_count | Countainer计数 | TEXT |
| object\_count | Object计数 | INTEGER |
| bytes\_used | 使用过的字节数 | INTEGER |
| hash | 数据db的hash值 | TEXT |
| id | 统一标识符 | TEXT |
| status | Account的删除状态 | TEXT |
| status\_changed\_at | 状态修改时间 | TEXT |
| metadata | account的元数据 | TEXT |

表4-2 container的数据项结构类型

|  |  |  |
| --- | --- | --- |
| Key | 含义 | 数据类型 |
| ROWID | 自增的主键 | INTEGER |
| name | Container的名称 | TEXT |
| object\_count | Container中object的个数 | INTEGER |
| bytes\_used | 使用字节数 | INTEGER |
| deleted | 是否标记为删除 | INTEGER |
| put\_timestamp | Put时间戳 | TEXT |
| delete\_timestamp | Delete时间戳 | TEXT |

表4-3 表incoming\_sync、表outcoming\_sync的数据项结构类型

|  |  |  |
| --- | --- | --- |
| Key | 含义 | 数据类型 |
| Remote | 远程节点的id | TEXT |
| Updated | 更新时间 | TEXT |
| Sync\_point | 上一次更新所在的行位置 | INTEGER |

Container中db文件和Account db相类似，具有相同的incoming\_sync表和outgoing\_sync表，另外还有三个新的表如表4-4、4-5所示分别是：与表account\_stat相类似的存放关于container信息的表container\_stat；记录object信息的object表，如下表所示：

表4-4 containrer\_stat的数据项结构类型

|  |  |  |
| --- | --- | --- |
| Key | 含义 | 数据类型 |
| created\_at | 创建时间 | TEXT |
| put\_timestamp | PUT请求指令的时间戳 | TEXT |
| delete\_timestamp | DELETE指令的时间戳 | TEXT |
| container\_count | Countainer计数 | TEXT |
| object\_count | Object计数 | INTEGER |
| bytes\_used | 使用过的字节数 | INTEGER |
| reported\_put\_timestamp | Report状态PUT的时间戳 | TEXT |
| reported\_delete\_timestamp | Report状态DEL的时间戳 | TEXT |
| reported\_object\_count | Report的object数 | INTEGER |
| reported\_bytes\_used | Report占用的字节数 | INTEGER |
| hash | 数据db的hash值 | TEXT |
| id | 统一标识符 | TEXT |
| status | Container的状态 | TEXT |
| status\_changed\_at | 状态修改时间 | TEXT |
| metadata | Container的元数据 | TEXT |
| x\_container\_sync\_point1 | 同步点1 | INTEGER |
| x\_container\_sync\_point2 | 同步点2 | INTEGER |

表4-5 object的数据项结构类型

|  |  |  |
| --- | --- | --- |
| Key | 含义 | 数据类型 |
| ROWID | 自增的主键 | INTEGER |
| name | Object的名字 | TEXT |
| size | object的大小 | INTEGER |
| creat\_at | 创建时间 | TEXT |
| Content\_type | 数据类型 | INTEGER |
| Deleted | Object的删除时间 | TEXT |
| etag | Delete时间戳 | TEXT |

#### 4.3.1.5 对象操作类

本文系统中设计存储系统中的对象进行修改、删除、搜索和更新操作时，设计实现上主要是用到ObjectController和DiskFile两个类来对文件进行操作，如图4-6所示，DiskFile类描述的是存储对象的属性信息，而类ObjectController是用来处理文件存储系统中的对象。



图4-6 对象操作类ObjectController和类DiskFile图

在ObjectController类中slow字段是用来控制操作请求的时间，async\_update()是用于在Object服务器发生变化，例如用户更新存储系统中对象container\_

update()或者对象被删除delete\_at\_update()，Object会发送HTTP请求Container更新容器数据，PUT/POST/HEAD/DELETE/GET等是用作具体的HTTP请求。而类DiskFile中metadata字段记录的是对象的元数据信息，meta\_file字段记录元数据信息的存储路径，put()可以将文件写入存储设备，drop\_cache()用作清空存储缓存。

### 4.3.2应用管理模块设计

应用管理模块是提供给服务提供商的应用管理员一个管理入口登陆管理门户，进行云存储系统日常正常运营和维护的操作平台，应用管理员可以进行用户权限的设置、查看系统租户情况、生成租户消费信息等操作。

#### 4.3.2.1租户明细查询

租户的具体请求操作(如PUT、GET、DELETE等)都会在系统产生的log文件中有

详细的记录，租户操作产生log日志结构如表4-7所示，系统可以根据用户请求查看上传到数据库(Mysql)中的log文件详细信息，因此本文系统提供给租户明细查询功能主要包括：查询客户端地址、操作请求方式和次数、请求结果、操作时间、使用空间、流入、流出流量信息。

表4-7 用户操作日志log结构

|  |  |  |
| --- | --- | --- |
| **字段** | **说明** | **示例** |
| Client\_IP | 请求发送的客户端地址IP | 192.168.1.50 |
| Remote\_Addr | 远程控制请求的IP地址 | 217.172.168.64 |
| Os\_group | 修改日志的用户所在组 | TestGroup |
| Log\_Time | 日志修改的时间 | Oct15 18:23:05 |
| RemoteAddr | 发出该请求的远端地址 | 192.168.1.51 |
| Strf\_Time | 请求收到的时刻 | 15/Oct/2013/18/23/05 |
| Request\_mothod | HTTP请求形式 | GET、LIST、DELETE、 PUT |
| Result | 返回的请求值 | 204(状态码) |
| Request | 请求的详细内容 | /v1/AUTH\_bcheje87ges8d7e98h3876d89ji8y5f |
| Protocol | 在请求中使用的协议 | HTTP/V1.0 |
| Referer | 参考引用的信息 | —— |
| UserAgent | 请求的用户代理信息 | Cyberduck/4.2.1%20%289350%29%20%28Windows%20XP/5.1%29%28x86%29 |
| AuthToken | 认证的令牌，Account名称以“%3A”符号截止，“%2C”前面是用户名、“%2C”显示用户验证码 | User\_13%3account3%2my\_cde545f eje87ges8d7e98h3876 |
| Send\_Id | 传送序列号 | Dfw7854dg45e7df245eg2qag2iu2e12yi2de |
| Flow\_InBytes | 发送请求的生成的上传流量 | 186 (Byte) |
| Flow\_OutBytes | 发送请求的生成的下载流量 | 263 (Byte) |
| Etag | 请求的HTTP头标签 | —— |

在实际操作上，租户对云存储系统发出的每一次请求(如GET、PUT、POST、PUSH、DELETE)通过代理服务器生成系统操作日志文件，本节设计主要用到系统操作日志文件以下字段：发出请求的客户端地址Client字段，请求方法ReqMothod字段、记录日志时间字段Log\_Time、写入日志的租户组及租户Os\_Group和Os\_User字段、请求权限令牌Auth\_Token字段、记录产生上传、下载流量的Flow\_InBytes和Flow\_OutBytes字段，请求操作具体信息的Request字段、返回请求结果的Result字段。

租户向系统请求查询操作明细要通过权限认证服务器的认证，其具体流程如下图4-7所示，租户向系统发出查询请求，代理服务器接收到租户请求并发送认证请求到权限认证服务器，此时认证服务器认证为真时，向系统发送请求处理租户内容(查询租户PUT、GET等操作)的请求，并通过代理服务器将结果反馈给租户，同样的没有通过权限认证代理服务器返回此次请求结果并更新记录本次请求，并生成新的日志文件记录租户详细操作情况。



图4-7 租户请求查询操作权限认证流程图

租户进行权限认证后，图4-8是对租户最近操作记录明细的打印流程，系统对日志文件随时监听，查看是否生成新的日志文件，如果发现租户最近一次生成的临时日志文件，读取日志文件并解析日志文件中的请求操作次数、操作上传流量、操作下载流量信息等，接着系统打印这些操作信息结果返回给租户，并将解析的数据信息生成一张租户操作数据信息表Statistics，同时会备份日志文件和数据库。



图4-8 打印租户操作信息流程图

#### 4.3.2.2租户权限设置

云存储服务提供商为公众客户提供安全、高速、大容量的在线存储管理、备份

等存储服务，在运营过程中，租户在一个计费周期内费用过期、超出使用费用等情况下，应用管理员就可以通过管理门户进行租户的冻结、删除或者恢复操作。

在设计实现上是通过认证服务器在租户进行身份认证的过程中，根据系统中租户的log信息进行查询，本文通过获得数据库中表account\_user的关键字段enable，可以进行租户的状态甚至，当enable值为真时恢复租户的账号，值为假时冻结租户的账号，实现租户权限管理功能。

#### 4.3.2.3租户消费计算

系统对租户进行费用结算的原理主要是按如下所述，按照一定的费用计算方

式，从数据库中读取租户的操作使用数据表Statistics，提取所需的数据信息按照计算公式计算。本文为了简化实验模型，设计使用了租户租用空间、操作请求次数及其产生流量、操作上传和下载流量几个要素进行简单的数学计算并进行展示，云存储运营方可以根据自己的业务发展模式提取更多租户操作信息数据，进行适合业务发展的费用结算。

在上文中表Statistics提取的租户操作信息包含的主要字段如下表4-8所示，当提取关心的字段数据再根据表4-9进行费用的清算，本文设计按照租户使用空间大小一天收费0.05元/GB，请求操作次数0.005元/每1000次，使用流量0.5元/GB，三部分相加形成用户的实际消费账单。

表4-8 表Statistics的数据结构

|  |  |  |
| --- | --- | --- |
| 字段 | 含义 | 数据类型 |
| ID | 主键，关联租户信息ID序号 | Intunsigned |
| ClientAddr | 租户IP地址 | varchar |
| AccountNanme | 请求操作的账号名称 | varchar |
| UserName | 请求操作的租户名称 | varchar |
| ReqTime | 操作请求时间 | datetime |
| ReqProtocol | 操作请求使用协议 | varchar |
| OutFlow | 输出流量 | Intunsigned |
| InFlow | 输入流量 | Int unsigned |

表4-9 系统设定计费公式表

|  |  |  |
| --- | --- | --- |
| 项目 | 计费标准 | 备注(计算单位) |
| 空间使用 | 0.02元 | GB |
| 操作使用流量 | 0.75元 | GB |
| GET、PUT命令请求次数 | 0.005元 | 每1000次 |
| 其它命令请求次数 | 0.0005元 | 每1000次 |

## 4.4 第三方接口模块开发与应用

### 4.4.1 应用需求分析

云存储具有通过使用其能力及强大的安全存储服务降低企业级存储成本的能力，通过对开源项目openstack的开发接口进行二次开发，企业及其开发者可以通过HTTP请求与本文云存储系统进行交互，实现文件的上传、获取图片、视频等媒体文件的功能。利用本文的云存储系统作为数据存储服务器，企业将WEB服务器的媒体文件等数据存储在云端即本文云存储系统中，将web服务与存储服务剥离开来，这样企业不需要购买昂贵的存储设备，减少企业运营成本。

传统的企业搭建WEB应用环境往往如图4-9所示的三层架构模型，将WEB应用处理和数据库搭建在WEB服务器上，WEB服务器的中间层WEB应用层接收来自互联网用户的HTTP请求，经过请求验证后由中间层将用户的请求翻译对后台的数据库进行查询、更新等获取数据的操作，并以可视化的界面在浏览器上展示给用户。传统的WEB应用环境下，WEB服务器不仅要处理来自用户的WEN应用请求与验证，企业庞大的数据信息存在WEB服务器上，加重了WEB服务器的负担，从而面临大量的WEB应用请求时，降低了WEB应用的处理速度。

图4-9 传统的WEB应用环境三层架构图

### 4.4.2网络存储应用架构

本文的云存储系统设计的第三方应用接口，企业或者开发者可以将WEB服务器上庞大的数据信息存储在云端，如下网络结构图4-10所示，云存储系统作为数据处理后台，不仅解决了安全验证问题，还有效的解决了数据存储、安全备份等问题，从而将企业中的WEB服务器的WEB应用请求处理和数据查询、更新的任务分开处理，WEB服务器只需要处理WEB应用的请求，不需要进行数据库的查询等工作，大大加快了WEB应用响应速度。从企业运营方面来说，将本文的云存储系统作为静态站点，处理后台的数据操作，企业只需要按需租用云服务的空间，而不在花费大量的成本在购置服务器硬件作为数据存储方面上，大大降低了运营成本，将更多的资金应用在运营维护、WEB应用优化等方面。

图4-10 第三方应用与云存储系统网络架构图

用户与企业的WEB应用服务器及数据存储系统(即本文云存储系统)的交互流程图如图4-11所示，首先，用户通过HTTP发送应用请求给WEB应用服务器，WEB应用服务器只需要处理用户的请求并返回操作结果给用户。接着，WEB服务器利用云存储系统设计的第三方应用接口申请的公用认证接口，通过云存储系统认证后，系统将用户请求应用操作数据直接发送给用户，整个过程中用户不需要知道请求数据是存储在云端，减轻了WEB服务器的负担，加快WEB应用请求的响应时间。



图4-11 第三方应用与云存储系统交互流程图

### 4.4.3权限认证（Authentication）

权限认证是由云存储系统提供的安全认证的过程，第三方应用进行权限的认证请求。本文设计的第三方认证规定，第三方访问向对象云存储系统发送REST请求时，在请求的HTTP头部必须包含X-Auth-Token这一特定的头部信息，此头部信息代表第三方的访问验证凭证“token”。REST接口中使用V1.0作为路径的GET请求方法，例如GET /v1.0 HTTP/1.1的请求方法。另外，验证的时候需要获取表示租户申请的用户名X-Auth-User和第三方接口访问密钥X-Auth-Key两个头信息值。

下图是第三方对云存储系统发送认证请求和响应的流程，租户发送第三方验证请求，并将第三方申请的用户名X-Auth-User和API接口访问密钥X-Auth-Key的两个值设置HTTP请求头的X-headers，请求验证方式如图4-12所示。

|  |
| --- |
| GET /v1.0 HTTP/1.1  Host: auth.cloudapi.test.com  X-Auth-User: test  X-Auth-Key: c78921cba3927gd2bc98546d62bb3d78 |

图4-12 第三方验证请求格式

验证服务器接到认证请求并处理请求，验证成功通过后，返回一个状态码为203的值(返回形式是2XX都是一个成功的响应)，如图4-13所示这个状态码包含了HTTP响应的头信息X-Storage-Url值为https://yourcloud.storage.com/v1/USER\_name\_34，还有X-Auth-Token值为bcheje87ges8d7e98h3876d89ji8y5f，第三方保存好这两个值并作为通过验证云存储系统的验证凭证。另外，本文系统还设计了状态码为401的错误请求页的相应，这状态返回一个额外的x-headers头信息告诉用户请求被拒绝。

|  |
| --- |
| HTTP/1.1 203 no content  Date: Tue, 15 Oct 2013 16:23:48 GMT  Server: Apache  X-Storage-Url: https://yourcloud.storage.com/v1/USER\_name\_34  X-Auth-Token: bcheje87ges8d7e98h3876d89ji8y5f  Content-Length: 0  Content-Type: text/plain; charset=UTF-8 |

图4-13 第三方验证请求响应图

### 4.4.4获取账号元数据

第三方成功通过验证后，用户就可以获取到云存储系统的数据，首先本文设计的账号服务接口可以让租户与云存储系统进行存储服务。在一个账号上，本文设计使用带有主机名称Host和验证令牌X-Auth-Token值的Head头部信息操作来获取账号下的容器Container的数量以及账号总存储字节数，如图4-14所示。相应的系统会返回两个系统定义的X-Account-Container-Count 和 X-Account-Bytes-Used的HTTP头标签值。

|  |
| --- |
| Head /<Api Version>/<Account> HTTP/1.1  Host: auth.api.test.com  X-Auth-Token: bcheje87ges8d7e98h3876d89ji8y5f |

图4-14 获取账号元数据请求

当Head头信息请求发送到账号存储服务器上时，账号服务器将处理Head头信息的请求，返回用户请求查看的信息，如图4-15所示用户使用Curl指令查看账户元数据信息，云存储系统将返回账户存储使用字节数X-Account-Bytes-Used值，账户中容器的数量X-Account-Container-Count以及对象的数量X-Account-Object-Count。

|  |
| --- |
| curl-k-H"X-Auth-Token:bcheje87ges8d7e98h3876d89ji8y5f"  https://192.168.1.47:8887/v1/AUTH\_ bcheje87ges8d7e98h3876d89ji8y5f  HTTP/1.1 204 No Content  X-Account-Object-Count: 0  X-Timestamp: 1265489756.68741  X-Account-Bytes-Used: 2300  X-Account-Container-Count: 3  Accept-Ranges: bytes  Content-Length: 12  X-Trans-Id: df985eb45dfd5256654fd65ddf52d4f9829ec  Date: Tue, 15 Oct 2013 17:23:48 GMT |

图4-15 获取账号元数据响应

### 4.4.5更新账号元数据

本文设计的系统让用户可以在云存储端account租户的级别上对自己的元数据进行更新，实际操作中也就是更新自己账号中的文件。设计上定义更新的元数据文件头以X-Account-Meta-\*形式创建或者更新，图4-16是用户创建一个新的元数据X-Account-Meta-File的请求格式。

|  |
| --- |
| Post /<Api Version>/<Account> HTTP/1.1  Host: auth.cloudapi.test.com  X-Auth-Token: bcheje87ges8d7e98h3876d89ji8y5f  X-Account-Meta-File: Michealhua  X-Account-Meta-Poto: Meta |

图4-16 请求更新账号元数据

带有更新元数据请求的Head发送到Account服务器上并请求处理，账号服务器将对Head头信息的作出反应，如图4-17所示系统会返回租户请求的响应，返回租户更新元数据X-Account-Meta-File的密钥Key的URL值，数据格式统一设计为utf-8。

|  |
| --- |
| curl –i -k -H "X-Auth-Token: bcheje87ges8d7e98h3876d89ji8y5f" -H X-Account-Meta-File-URL-Key:9kjd9ie56984kjde89dc23fd79846d9c https://192.168.1.50:8080/v2/AUTH\_bcheje87ges8d7e98h3876d89ji8y5f  HTTP/1.0 203 No Content  Content-Type: text/html; charset=utf-8  Content-Length: 2  X-Trans-Id: df985eb45dfd5256654fd65ddf52d4f9829ec  Date: Tue, 15 Oct 2013 17:25:30 GMT |

图4-17 更新元数据响应交互图

### 4.4.6查询容器列表

本文设计对一个租户的账号的X-Storage-Url进行GET的方式来获取这个账号的容器列表信息，并且按照名称的顺序进行排序，这个排序的程序是按照二进制决定的SQLite的内建排序的方法memcmp()函数来实现的，下表4-10是函数定义支持请求的可选参数：

表4-10 memcmp( )函数数据类型表

|  |  |  |
| --- | --- | --- |
| 参数 | 含义 | 数据类型 |
| limit | 返回结果的个数 | 整数n |
| format\_type | 指定为XML或JSON数据格式的响应 | xml或json |
| end\_marker | 返回名称比指定字符串小的container列表 | 字符串x |
| marker | 返回名称比指定字符串大的container列表 | 字符串x |

租户test使用申请的账号X-Storage-Url使用GET的方式请求格式如图4-18所示，租户向账号服务器提供令牌Auth\_Token，并发送列出容器元数据的信息请求。

|  |
| --- |
| Get /api v\_1.0/account=test HTTP/1.1  Host: auth.cloudapi.test.com  X-Auth-Token: bcheje87ges8d7e98h3876d89ji8y5f |

图4-18 请求列出容器信息

本文设计的云存储系统都是通过代理服务器(Proxy Server) 向外提供基于 HTTP 的 REST 服务接口，用户可以对帐号以及帐号中的容器、对象进行GET、CURL等请求操作，但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，就像是上述的X-Auth-Token值“bcheje87ges8d7e98h3876d89ji8y5f”。

账号服务器收到租户请求列出容器信息的请求后，根据租户的请求查看容器信息按照SQLite的内建排序的方法memcmp()函数，如图4-19所示，系统将会列出账户中容器含有容器apple、backup-file等8个容器。

|  |
| --- |
| curl -k –i -X GET -H  "X-Auth-Token: bcheje87ges8d7e98h3876d89ji8y5f "  https://192.168.1.50:8080/v1/AUTH\_ bcheje87ges8d7e98h3876d89ji8y5f  HTTP/1.1 202 OK //返回HTTP的状态码  X-Account-Object-Count: 0 //账户中对象的数量  X-Timestamp: 1265489756.68741 //时间戳  X-Account-Bytes-Used: 273 //账户使用的字节数  X-Account-Container-Count: 8 //账户中容器的数量  Accept-Ranges: bytes  Content-Length: 55  Content-Type: text; charset=utf-8 //数据文本格式  X-Trans-Id: df985eb45dfd5256654fd65ddf52d4f9829ec  Date: Tue, 15 Oct 2013 17:33:28 GMT //打印时间  apple //容器列表  backupfile  docfile  file  girls  image  music  testfile |

图4-19 列出容器信息交互图

本文设计获取账户容器时，考虑到企业账户的容器个数往往是比较大，所以在memcmp()函数中使用两个参数进行控制，分别是参数limit和参数marker，参数limit控制列出账户中容器的个数，而参数marker列出那些比用户指定字符串较大的参数的

容器名称，如图4-20所示指定列出字符串比“fire”大的4个容器名称。

|  |
| --- |
| curl -k –i -X GET -H  "X-Auth-Token: bcheje87ges8d7e98h3876d89ji8y5f "  https://192.168.1.50:8080/v1/AUTH\_ bcheje87ges8d7e98h3876d89ji8y5f?marker=fire\&limit=4  HTTP/1.1 202 OK //返回HTTP的状态码  X-Account-Object-Count: 0 //账户中对象的数量  X-Timestamp: 1265489756.68741 //时间戳  X-Account-Bytes-Used: 273 //账户使用的字节数  X-Account-Container-Count: 8 //账户中容器的数量  Accept-Ranges: bytes  Content-Length: 55  Content-Type: text; charset=utf-8 //数据文本格式  X-Trans-Id: df985eb45dfd5256654fd65ddf52d4f9829ec  Date: Tue, 15 Oct 2013 17:39:28 GMT //打印时间  girls //容器列表  image  music  testfile |

图4-20 列出比指定字符串大的容器列表图

另外，第三方在编写应用的时候，或者需要以XML或者是JSON的形式获取容器信息，本文设计以参数format\_type来控制返回给用户的数据类型，例如设置format\_type=json，其返回的数据格式如图4-21所示。

|  |
| --- |
| [{"name":"apple","count":1,"bytes":273},{"name":"backupfile","count":2,"bytes":834},{"name":"docfile","count":3,"bytes":9261},{"name":"file","count":4,"bytes":73},{"name":"girls","count":5,"bytes":563},{"name":"image","count":6,"bytes":1656},{"name":"music","count":7,"bytes":454}{"name":"testfile","count":8,"bytes":42425}] |

图4-21 JSON形式返回数据图

## 4.5本章小结

本章是云存储系统详细设计的主要部分，详细描述了分布式对象云存储系统的整体设计流程。首先，根据云存储系统的功能需求，提出了一个高可用、可弹性拓展的云存储系统框架；其次，分别描述了本文云存储系统内容管理存储模块、应用管理模块的设计思路和设计流程，设计了系统操作日志的数据格式、三类存储服务器的账号、容器以及对象的数据表格式、用于统计分析数据库表Static格式；最后，阐述了查询租户操作记录、统计分析、费用计算功能模块的设计流程和方法。

另外，本文设计并提供第三方接口实现第三方应用与云存储系统的交互，将本文的云存储系统作为企业WEB应用的数据存储站，减轻了WEB服务器的负担，加快WEB应用请求的响应时间。

# 第五章 云存储平台及第三方应用测试

## 5.1 云存储系统部署环境说明

本文的云存储系统的网络结构如下图5-1所示，代理服务器、认证服务器和存储服务器集群组成本文的云存储应用系统，在下面小节中，将依次进行云存储系统的内容管理功能应用测试、应用管理功能测试和WEB应用服务器使用本文设计的第三方接口与云存储系统交互的测试，将传统的WEB服务器上的媒体文件上传到本文的云存储系统，并展示WEB服务器发出HTTP请求从云存储系统下载文件的整个过程，下面介绍一下本文云存储系统使用到的硬件环境和软件环境：

(1)硬件环境：

6台服务器；分别作为代理节点、认证节点和存储节点，用户通过公网IP地址进行访问租户身份验证，验证通过后通过可视化界面Horizon进行存储服务，本文设置的公网IP地址是192.168.1.50，存储节点内网IP为10.0.0.X，监听端口为9002。

(2)软件环境

操作系统：linux环境下的操作系统Ubuntu Server版ubuntu-12.10-server-amd64

应用软件：linux环境下的python环境、C环境、Django WEB框架、Ubuntu远程ssh登陆SecureCRT 7.0、ssh网络映射SftpDrive 1.7.9



图5-1 云存储系统网络结构图

## 5.1 内容管理存储模块测试

### 5.1.1 租户注册功能测试

图5-2是云存储系统的登陆首页界面，用户可以点击创建用户按钮，填写相关信息注册用户名、邮箱及密码向服务提供商申请租户“test”身份，享受云存储服务。



图5-2 云存储系统注册界面图

### 5.1.2 基本存储功能测试

本小节测试的是租户通过身份验证登陆云存储系统进行基本的存储服务，租户登陆云存储系统，在自己的存储空间新建一个“论文终稿”的新容器如图5-3所示，同时租户删除容器容器操作如图5-4所示，租户上传本地文件“desert.jpg”到新建的容器“论文终稿”中，如图5-5、5-6所示。



图5-3 创建新的容器测试图



图5-4 容器删除操作测试图



图5-5 上传文件对象操作测试图



图5-6 选择本地文件上传测试图

### 5.1.3 查看账户明细

租户在自己账户界面参看自己使用云存储服务使用情况，如图5-7所示，租户test1查看到自己使用了20G的存储空间，按照设计中定义每1GB存储空间收费0.02元，所以空间使用费用为0.4元；相同的租户在存储操作时候使用流量为2GB，按照1GB产生流量费用为0.75元计算，流量产生费用为1.5元；以及请求GET操作次数没有超过1000次因此不用计费，故租户需要付费1.9元。

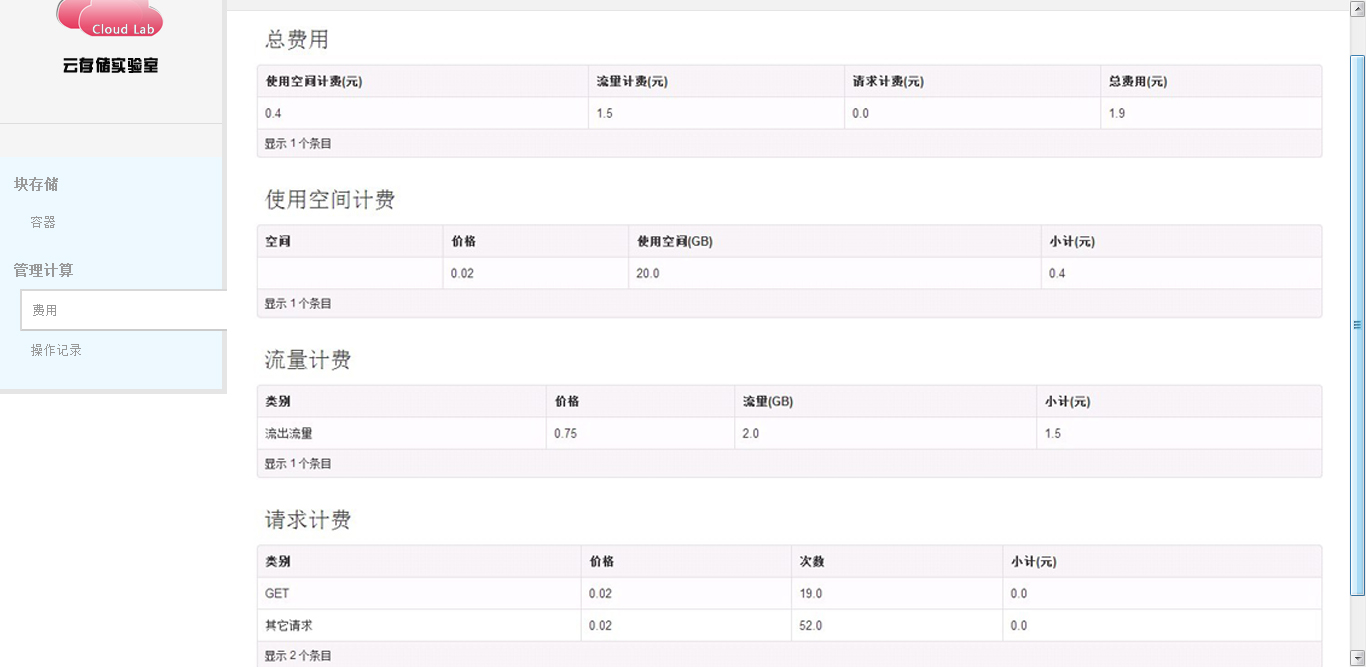


图5-7 查看账户清单测试图

## 5.2 应用管理功能模块测试

本节主要测试了云存储系统管理员查看系统中租户使用情况、查看租户消费情况、对租户权限管理操作三个方面的功能测试。

### 5.2.1 查看系统租户情况功能测试

云存储系统应用管理员admin1通过权限认证，登陆应用管理平台，由图5-8可以看到应用管理员使用管理员身份进行查看系统中租户的注册的租户user1、user2、user3等租户的名称、ID号、注册邮箱、激活状态信息。

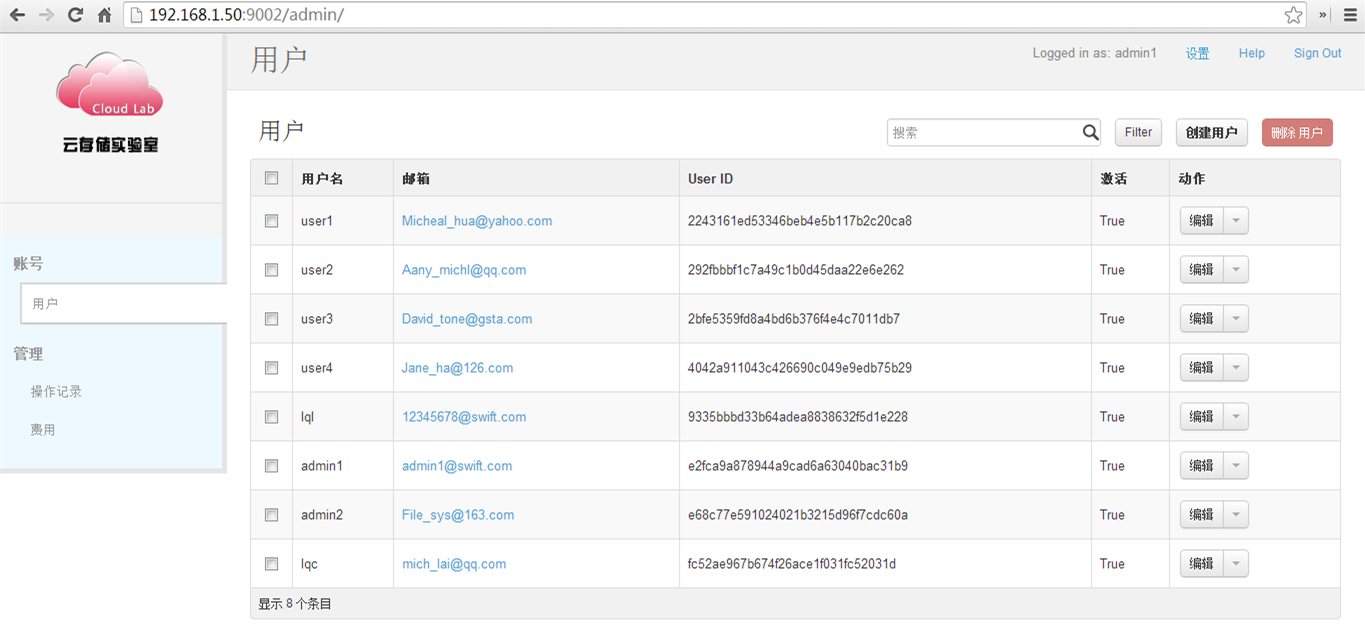


图5-8 查看系统租户列表测试图

如图5-9所示，管理员admin1查看租户test1在 10月1日至18日期间，流出流量107M、GET次数50次，POST次数为9次等具体操作信息。

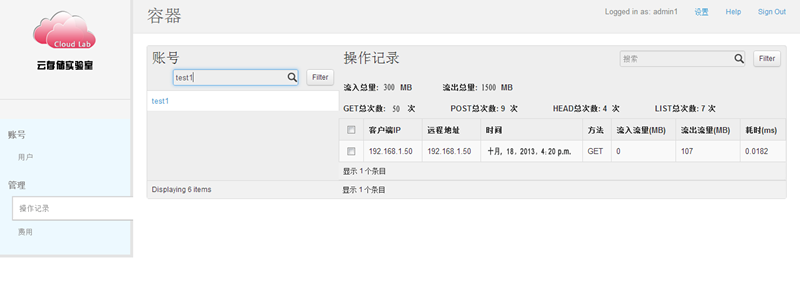


图5-9 查看租户操作记录测试图

### 5.2.2 查看租户消费情况测试

由图5-10所示，应用管理员对租户tet1的消费情况进行查询，租户test1使用空间费用为0.4元、流量使用费用为1.5元，并没有产生请求操作PUT/GET等费用，租户总共需要缴纳费用为1.9元，账户余额还剩下60元。



图5-10 查看租户费用信息测试图

### 5.2.3 管理租户权限测试

对于过期没有为云存储服务付费的用户，服务提供商可以对其进行冻结、禁用和删除操作，图5-11所示应用管理员admin1冻结租户user1的云存储服务使用权限，图5-12所示管理员删除用户user1，结束对租户user1的服务并释放租户存储资源。

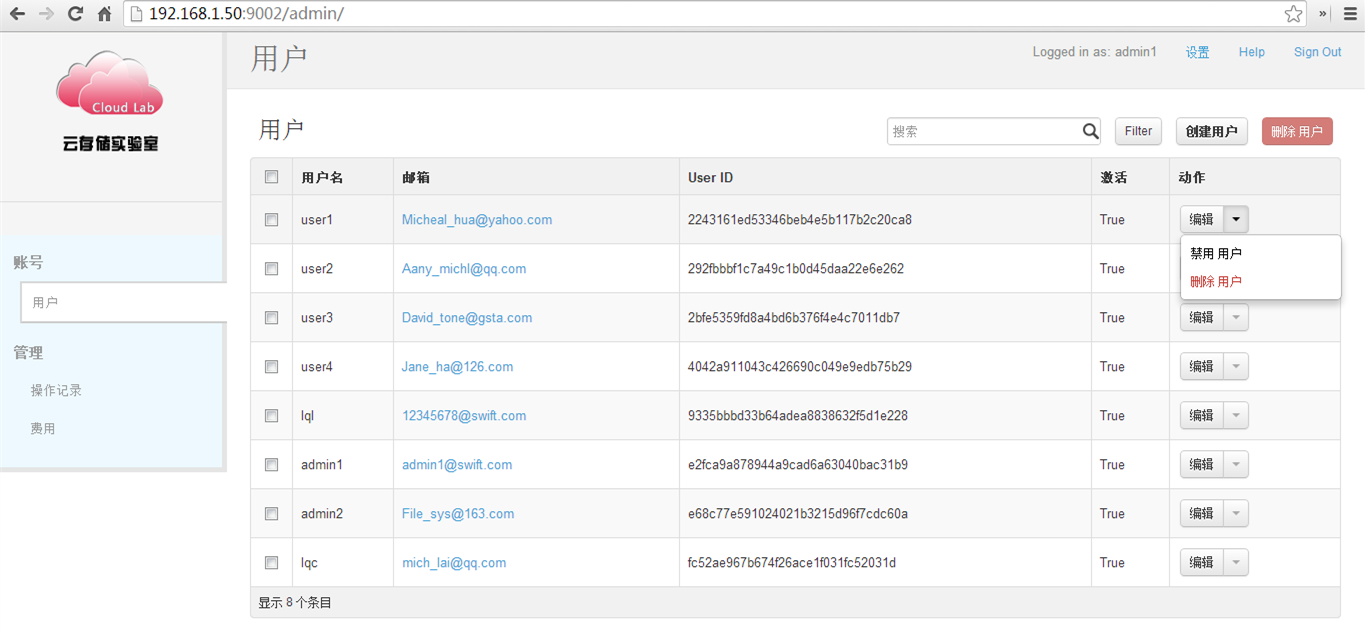


图5-11 冻结、删除租户账号测试图

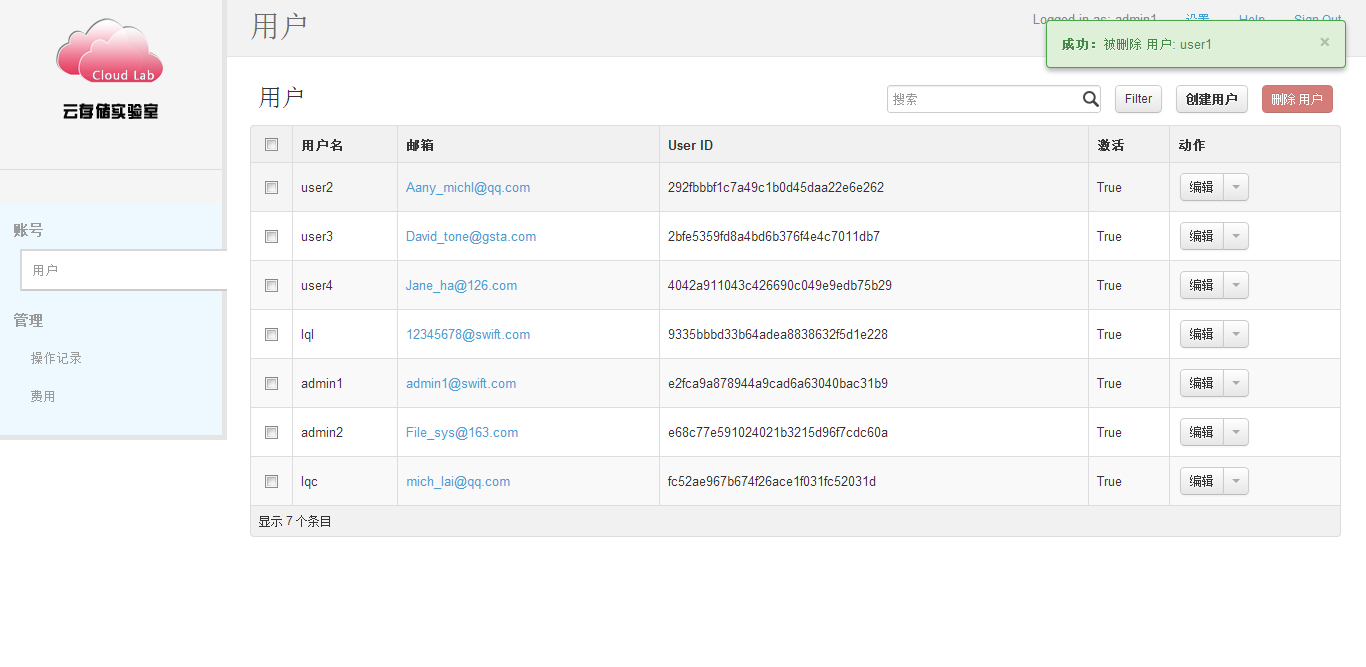


图5-12 删除租户账号测试图

### 5.2.4 运维管理测试

如图5-13所示，应用管理员查看到云存储系统的存储服务器CPU为10个、系统MEM内存为20G、存储容量为500G，决定是否对系统进行弹性拓展，增加存储节点，扩大云存储系统规模。



图5-13 查看云存储系统信息测试图

## 5.3 第三方应用测试

### 5.3.1 第三方应用测试说明

本小节将使用第三方开源工具WordPress制作一个云存储项目的展示网站，将云存储系统做为数据处理后台，利用本文设计的第三方应用接口实现WEB应用请求和查询处理数据库任务的分离，减轻了WEB服务器的负担，加快WEB应用请求的响应时间。具体测试的环境部署网络示意图如图5-14所示，WEB服务器将多媒体文件等数据存储在云存储系统中，WEB服务器只需要处理WEB应用的请求，不需要进行数据库的查询等工作，大大加快了WEB应用响应速度。

图5-14 第三方应用测试网络结构图

### 5.3.2 WEB端网页上传功能测试

用户在WordPress制作的网页端如图5-15所示，选择本地多个图片文件上传到WEB服务器，实际上是WEB服务器将用户数据存储到本文的云存储系统中，图5-16是用户admin上传本地文件“1.jpg”图片文件到WEB服务器上。



图5-15 WEB网页端首页界面图

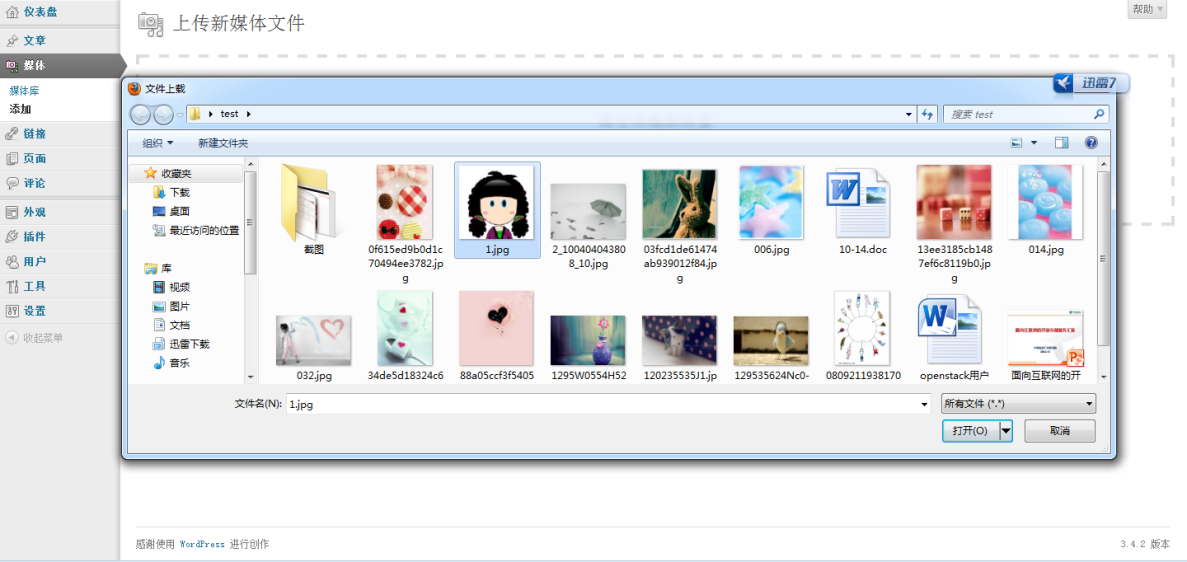


图5-16 上传本地文件到WEB端示意图



图5-17 WEB网页端显示文件列表图

### 5.3.3云存储端功能测试

由图5-18可以看到，云存储和第三应用方通过认证登陆云存储系统后台可以看见用户刚刚上传的“1.JPG”图片文件，实现了WEB服务器的处理WEB应用请求和存储数据的分离，图片、音频、视频等多媒体文件实际上传到云存储后台进行存储。

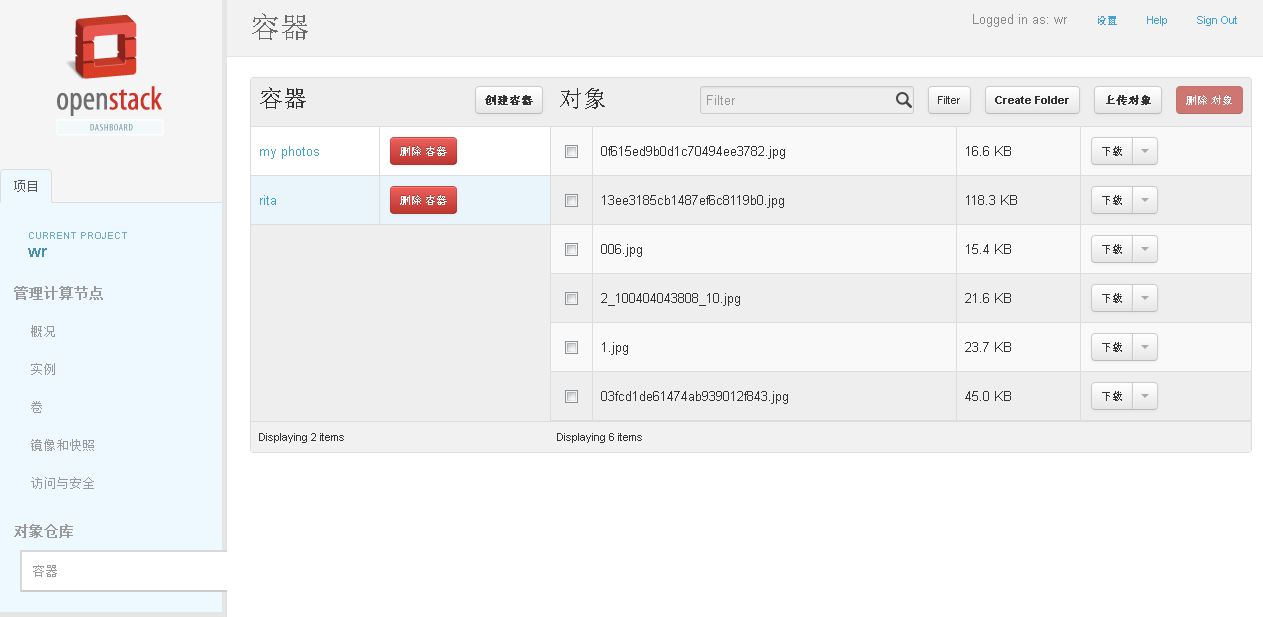


图5-18 云存储端显示用户文件列表图

### 5.3.4 WEB端删除文件功能测试

由图5-19、5-20所示，用户admin在WEB网站端删除图片文件“1.JPG”的操作，相应的云存储管理员可以看到云存储端的刚上传的图片文件“1.JPG”也相应的被删除，如图5-21所示：



图5-19 WEB网页端删除文件图

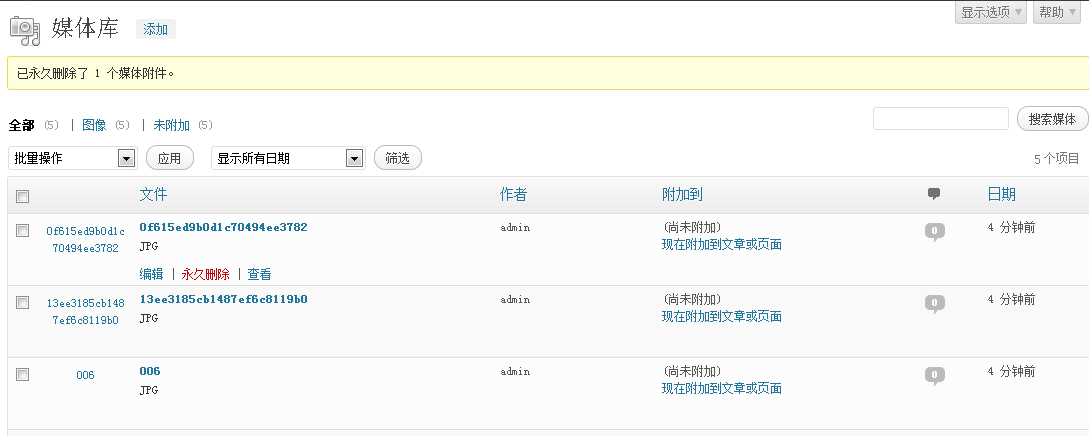


图5-20 WEB端删除文件后列表示意图

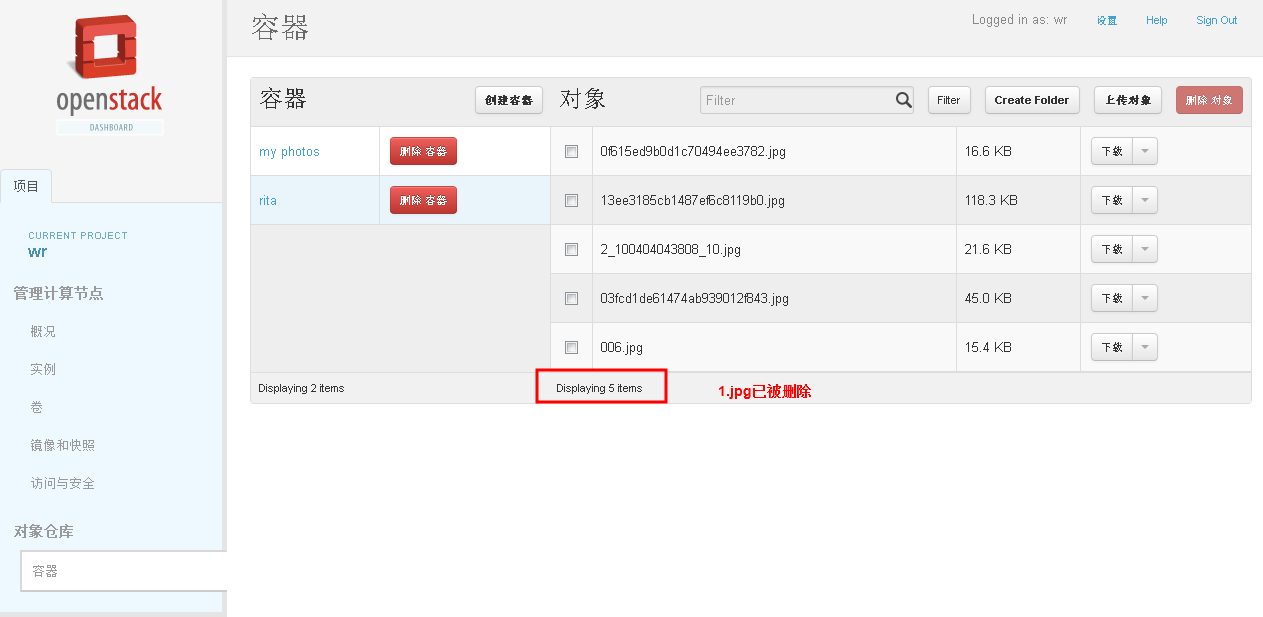


图5-21 云端存储系统显示用户文件列表图

## 5.4 本章小结

本章主要是对本文云存储系统的基本云存储服务功能和业务应用管理功能模块的测试，并用WordPress模拟第三方应用将云存储系统作为数据处理平台，进行第三方的云存储服务功能测试。测试表明，本文云存储系统提供高效可用的存储服务，并通过界面风格友好、操作性简单可行的可视化界面Horizon进行业务应用管理，实现查看租户详情、费用使用清单等操作。最后通过第三方应用WordPress，使用本文云存储系统做为WEB服务器的后台数据存储平台，有效的将数据存储和WEB应用请求分离开来，解决了传统WEB应用部署中服务器因请求过大造成响应延迟问题，并从另一方面也验证了本文云存储系统的可用性。

# 总结与展望

2013年是“云计算的实践与实战元年”，从2012年开始，国内各大厂商看到了国外运用云计算成功进行业务转型和增长的典型案例，纷纷加入云计算的战略发展和部署。而在这个发展机遇面前，中国电信作为全球领先的服务提供商，拥有丰富的带宽资源和庞大企业及公众用户数量资源，在告诉发展的云计算领域更需要整合自身资源，对内需要进行海量的数据存储、管理以及业务分析的需要，对外也可以提供存储服务接口提供高可用、可弹性拓展的云存储服务。因此，基于上述的情况和需求，本文旨在提出一个可行的云存储服务系统框架，设计并实现提供基础存储服务的云存储平台和适合业务管理的应用管理平台，设计并开发第三方应用接口对外提供存储服务。

首先，本文深入分析了三种传统网络存储技术：直连存储DAS、联网存储NAS和存储区域网SAN，分析了他们的存储网络结构模型、存储机制原理和技术特点，对比分析三种存储网络架构特征，分析得出传统存储技术存在的不足与不适应新的数据存储技术的需求，从而引出新型的对象存储技术。其次，通过深入研究对象的网络结构模型和基础功能模块，提出一个满足课题需求的高可用、可弹性拓展的云存储系统框架，主要包括提供基础云存储功能的内容管理模块和进行业务管理的应用管理模块。

再次，本文详细的阐述了系统功能模块的设计流程和方法。在实际设计中，基础云存储服务平台使用了开源项目Openstack存储的基本架构，设计了三类存储服务器的数据表格式、存储路径生成及对象搜索映射机制的基本流程；Openstack存储架构没有涉及业务管理的应用管理平台，本文设计并实现一个业务运营的应用管理功能模块，主要包括查看租户情况、查询租户操作详情、费用结算、系统使用情况等功能的运维及应用管理平台。

最后，本文还对云存储系统的各个功能模块进行功能测试，测试表明本文云存储系统拥有云存储基础存储服务功能，并提供一个界面良好、操作简单的Horizon可视化界面提供给服务提供商进行业务管理和系统运维监管。另外，设计并开发第三方应用接口功能，解决传统的WEB应用服务的应用服务处理和数据处理加重服务器负担，降低响应速度的难题。

通过实验测试说明云存储系统具备高可用云存储服务能力、符合运营商和服务提供商进行业务管理的应用管理功能、能够提供第三方应用的开放存储功能。

当然，此系统在许多方面还有待改进和完善，本文提出以下几点作为日后深入研究的方向：

1、本文的存储系统由于资源有限，还未进行在大规模数据处理情况下的压力测试，系统的稳定性能还有待提高。

2、现在的PC、平板电脑及移动设备广泛普及与应用，下一步可以设计并实现多平台支持、跨平台操作的云存储服务。

3、在第三方应用上以及云存储认证的安全机制还有待改进，可以进行多维度的身份认证例如密钥、账户密码、安全锁认证等。

# 参考文献

1. Bhardwaj S, Jain L, Jain S. Cloud computing: A study of infrastructure as a service (IAAS)[J]. International Journal of engineering and information Technology, 2010, 2(1): 60-63.
2. 许四平. SaaS 软件即服务模型研究[J]. 硅谷, 2009 (004).
3. 吴吉义, 平玲娣, 潘雪增, 等. 云计算: 从概念到平台[J]. 电信科学, 2009, 12(11).
4. 郭本俊, 王鹏, 陈高云, 等. 基于 MPI 的云计算模型[J]. Computer Engineering, 2009, 35(24).
5. 李智. 下一代通信机房及 IDC 数据中心能源成本战略[J]. 第十届中国科协年会信息化与社会发展学术讨论会分会场论文集, 2008: 458-462.
6. 廖军, 谭浩. 新一代开放分布式处理技术——Web Services[J]. 计算机应用, 2004, 24(8): 5-9.
7. 苏光大. 图像并行处理技术[M]. 清华大学出版社, 2002.
8. 徐志伟, 冯百明, 李伟. 网格计算技术[M]. 电子工业出版社, 2004.
9. 王耀南. 计算智能信息处理技术及其应用[M]. 湖南大学出版社, 1999.
10. 曾碧卿, 陈志刚. 服务器集群系统研究[J]. 计算机应用研究, 2004, 21(3): 186-187.
11. 曹培发. 集群文件系统数据服务器研究[J]. 北京: 装备指挥技术学院, 2005.
12. Gibson G A, Van Meter R. Network attached storage architecture[J]. Communications of the ACM, 2000, 43(11): 37-45.
13. Liu C Y, Chen H B. Network attached storage: U.S. Patent D563,994[P]. 2008-3-11.
14. 蔡皖东. 基于 SAN 的高可用性网络存储解决方案[J]. 小型微型计算机系统, 2001, 22(3): 284-287.
15. 余胜生, 赵玉峰, 周敬利. Fibre Channel 主机适配器的研究与设计[J]. 小型微型计算机系统, 2002, 23(6): 663-666.
16. 陈凯, 白英彩. 网络存储技术及发展趋势[J]. 电子学报, 2002, 30(12): 1928-1932.
17. 李明琪. SCSI 总线[J]. 微处理机, 1992 (4): 55-61.
18. 冯丹. 网络存储关键技术的研究及进展[J]. 移动通信, 2009, 33(11): 35-39.
19. 刘群, 冯丹, 王芳. 高可靠性元数据服务器研究[J]. Computer Engineering, 2008, 34(17).
20. 冯丹, 史伟, 覃灵军, 等. 基于对象存储系统的对象文件系统设计[J]. 华中科技大学学报: 自然科学版, 2006, 34(12): 47-49.
21. Burns A, Wellings A J. Real Time Systems and Their Programming Languages: Ada 95, Real-time Java and Real-time POSIX[M]. Pearson Education, 2001.
22. 李世畅, 杨浩澜, 李世亚, 等. NAS 存储技术的研究与应用[J]. 计算机工程与应用, 2003, 13: 194-196.
23. 魏青松. 大规模分布式存储技术研究 [D][D]. 成都: 电子科技大学博士学位论文, 2004.
24. 张顺达. 对象存储系统的元数据管理[D]. 武汉: 华中科技大学, 2006.
25. 史小冬. 分布式文件系统高可用问题研究[J]. 博士学位论文, 中科院计算技术研究所, 2002.
26. 方圆. 基于对象存储元数据管理策略的研究与实现[D]. 解放军信息工程大学, 2012.
27. 付东华. 基于 HDFS 的海量分布式文件系统的研究与优化[D]. 北京邮电大学, 2012.
28. 冯丹. 网络存储关键技术的研究及进展[J]. 移动通信, 2009, 33(11): 35-39.
29. 吴吉义. 基于 DHT 的开放对等云存储服务系统研究[D]. 浙江大学, 2011.

# 攻读硕士学位期间取得的研究成果

一、已发表（包括已接受待发表）的论文，以及已投稿、或已成文打算投稿、或拟成文投稿的论文情况**（只填写与学位论文内容相关的部分）：**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **作者（全体作者，按顺序排列）** | **题目** | **发表或投稿刊物名称、级别** | **发表的卷期、年月、页码** | **相当于学位论文的哪一部分（章、节）** | **被索引收录情况** |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

注：在“发表的卷期、年月、页码”栏：

1如果论文已发表，请填写发表的卷期、年月、页码；

2如果论文已被接受，填写将要发表的卷期、年月；

3以上都不是，请据实填写“已投稿”，“拟投稿”。

不够请另加页。

二、与学位内容相关的其它成果（包括专利、著作、获奖项目等）

在硕士期间，本人主要参与并完成项目如下：

1. 云计算CDN流媒体分发平台研发 项目开发时间：2011-09-01至2011-12-31

企事业单位委托技术开发项目 合作单位：中国电信股份有限公司广东研究院

1. 基于CDN的可伸缩视频分发理论与技术研究 2012-01-01至2013-12-31

# 致谢

研究生生活在不知不觉中悄然的度过，在这两年半的研究生期间不仅让我学习到丰富的知识，还收获很多真挚的友谊。尽管在科研和生活中都会有许多困难和挑战，在克服困难的同时，也让我慢慢的成长，学习如何去面对压力迎接挑战，在这段精彩的日子里，我非常感谢可以和我一起成长的同学、老师和朋友们。

首先，我要感谢智能信息处理实验室的领头人，谢胜利教授；因为有您辛勤的劳动，才有实验室辉煌的今天，让我们整个团队在各位导师的带领下可以专心科研，为追求科学真理而拼搏奋斗。

其次，我要感谢我的导师吴宗泽老师。在科研学习上，你严谨的治学态度，时刻都鞭策着我努力向着目标前进；在平常的生活中，你是我们的老大，对我们生活的点点滴滴关怀备至，无论何时都会把您的学生放在重要的位置，你是我们永远的吴老大。

同时，也感谢我们实验室的傅予力老师、周智恒老师、向友君老师、李波老师、余荣老师、谭北海老师等各个指导老师，在平时的科研生活中有你们的教导，让我慢慢的成长。

在生活中也有一帮可爱的小伙伴陪着我度过这段美好的时光，当然就是在同一个实验室一起奋斗的可爱朋友们，吴容、李爽、曾星、耀程、啓程、耀波、罗涛、伊竹、石清、国监、妍蓉、老蒋等等，有你们陪伴和鼓励，让我的生活更加精彩和美丽。我的室友们国贤、杨烨、陈博，感谢你们给我难忘的宿舍生活，让我有一段难忘的回忆。

最后，我要特别感谢我的父母，有你们的教导和悉心栽培才有今天成就的我，是你们用血汗换来我今天的所有一切，在这个即将结束的研究生生涯，我想和你们说一声：“我永远深深爱着你们，谢谢爸爸、妈妈、妹妹以及所有爱我的人”。

感谢在论文期间，各位参与开源Openstack讨论的牛人们，谢谢你们的指导。

感谢审查论文的老师们，祝您们工作顺利，家庭幸福。

赖文华

2013年10月于广州华南理工大学

