

华东师范大学

East China Normal University

本科生毕业论文

基于 Linux 单片机树莓派的云盘系
统的设计与实现

Design and Implementation of Cloud
Storage System Based on Linux Single
Chiptop Raspberry Pi

姓 名: 吴清泽

学 号: 10152510231

学 院: 计算机科学与软件工程学院

专 业: 软件工程

指导教师: 沈佳辰

职 称: 工程师

2019 年 6 月

目录

摘要	I
Abstract	II
1. 绪论	1
1.1 研究背景	2
1.2 本文贡献	5
1.3 研究思路	8
1.4 论文组织	9
2. 需求分析	10
2.1 系统总体需求分析	10
2.2 功能性需求分析	11
2.3 非功能性需求分析	14
2.4 系统可行性分析	15
3. 系统设计	17
3.1 系统设计目标和原则	17
3.2 服务器设计	17
3.3 数据库设计	19
4. 系统实现	23
4.1 系统实现环境	23
4.2 系统功能实现	23
5. 系统测试	32
5.1 测试方法	32
5.2 测试环境和测试工具	34
5.3 系统功能测试	35
6. 总结和展望	39
6.1 研究和工作总结	39
6.2 展望	39
参考文献	42
致谢	43

摘要

信息科技的广泛应用带来了互联网经济的飞速成长，我们的社会也进入了大数据时代，每个使用互联网的个人始终面临着数据的存储访问的问题。十几年前困于宽带速度和硬件成本的限制，个人用户在数据存储、传播、访问、备份、同步、分享等方面的需求都不能得到充分的满足。随着硬件成本和网络资费的下降，人们越来越能够高效、安全地存储和处理海量数据，同时兼顾成本和部署调整的灵活性和速度。人们可以利用云存储技术实现具备多终端数据同步、平台无缝连接、资源共享等功能的个人云存储应用。

本文首先分析了市场上主流的云盘服务的市场定价，发现这些服务的定价对于普通用户来说还是太昂贵了，因此，本人设计了一种基于 Linux 单片机树莓派的云盘系统，并开发了 web 客户端和安卓客户端，旨在为用户提供一种廉价的云存储服务。web 客户端和安卓客户端将服务端文件存储目录映射为本地的目录索引，用户可以如同操作本地文件一样操作服务端文件，例如服务器文件的增删改查操作。本系统采用典型的 MVC 体系架构设计实现各逻辑层之间的数据通信，为用户提供文件的上传、下载、分享、搜索、备份等功能。

在完成需求分析和系统实现的工作之后，本文对系统的用户管理、系统配置、文件操作 3 个方面进行功能性测试，同时对本地、服务器的磁盘读写和网络速度和稳定性做了性能测试。实验结果表明，本云盘系统在对 3、4 个并发用户的并发场景下文件读写吞吐率可以达到约为 75Mb/S，基本上满足系统开始的设计目标，同时本系统提供界面美观的客户端，用户体验良好。

关键词：网盘，云盘，云存储，私人网盘，安卓，树莓派

Abstract

The widespread use of information technology has brought about the rapid growth of the Internet economy. Our society has also entered the era of big data. Everyone who uses the Internet has always faced the problem of data storage access. More than a decade ago, due to the limitations of broadband speed and hardware cost, individual users' needs for data storage, transmission, access, backup, synchronization, and sharing could not be fully met. As hardware costs and network tariffs decline, people are increasingly able to store and process massive amounts of data efficiently and securely, while at the same time taking into account the flexibility and speed of cost and deployment adjustments. People can use cloud storage technology to implement personal cloud storage applications with multi-terminal data synchronization, seamless platform connectivity, and resource sharing.

This paper first analyzes the market pricing of mainstream cloud disk services in the market, and finds that the pricing of these services is still too expensive for ordinary users. Therefore, I designed a cloud disk system based on Linux MCU Raspberry Pi and developed it. The web client and Android client are designed to provide users with an inexpensive cloud storage service. The web client and the Android client map the server file storage directory to the local directory index, and the user can operate the server file as if the local file is operated, such as the addition, deletion, and modification of the server file. The system adopts the typical MVC architecture design to realize data communication between each logical layer, and provides users with functions such as uploading, downloading, sharing, searching and backuping files.

After completing the requirements analysis and system implementation work, this paper conducts functional tests on the user management, system configuration, and file operations of the system, and performs performance tests on local and server disk read and write and network speed and stability. The test result analysis shows that the cloud disk system has controllable storage resource consumption of the system, and its file read/write throughput rate is about 75 Mb/s, which fully meets the needs of users and has a good user experience.

Keywords: cloud storage, cloud storage, private network storage, Android, Raspberry Pi

1. 绪论

随着计算机技术和网络技术的发展，人们将计算机和网络应用在自己的工作学习娱乐生活中的场景越来越高，尤其是在流量资费下降和移动手机、个人计算机日渐普及的社会背景下，人们可以轻易的制造、获取、传播文件资源，在这一过程中，有限的终端存储空间越来越不能满足人们日益增的存储需求，而且终端设备具有强大的移动属性，因此数据损坏的风险也不能够忽视^[1-3]。因此本文设计开发了一套云端的存储系统，可以帮助人们解决移动终端存储空间不足和数据损坏丢失风险的问题。

本系统是利用闲置的家庭宽带和淘汰的硬盘资源，在基于廉价的通用小型计算机树莓派（英语：Raspberry Pi）的基础上开发出一个提供文件的备份、共享、存储、访问、搜索、管理等功能，满足多终端使用的在线云盘存储系统。具体的设计与实现方案是将后台服务部署到树莓派单片机上，同时开发出 Web 和安卓 APP 客户端来备份和访问文件。技术工具基于 Apache/2.4.25 (Raspbian), MariaDB (MySQL 的树莓派版本) 10.1.37, Android, PHP 7.0.33, jquery, jquery-ui, bootstrap 等^[4, 5]。

Raspberry Pi(中文名为“树莓派”，简写为 RPi, (或者 RasPi / RPI) 是为学习计算机编程教育而设计，只有信用卡大小的微型电脑，其系统基于 Linux。随着 Windows 10 IoT 的发布，树莓派也将可以用上运行 Windows 的树莓派。自问世以来，受众多计算机发烧友和创客的追捧。树莓派单片机在市场上的定价为 35 美元，所以相比于租用阿里云、腾讯云等等云服务来部署云盘系统，树莓派性价比笔更高，这也是本文基于树莓派来开发云盘系统的原因。虽然树莓派的性能不是很高，但是本系统是一个 I/O 任务远大于计算的项目，而且定位是一个私人的网盘系统，所以对并发和计算的要求不高，基本上满足同寝室四五个人的同时上传下载访问就可以了。而且本项目的使用场景大多数是在寝室的局域网里面，所以对网络也没有做的太多的优化，其实最好的网络优化算法都比不上一个最低的丢包率，因为本云盘系统是部署在局域网里面的，所以在寝室的局域网的网络条件下丢包率非常小^[4, 6-11]。另外虽然本系统是部署在局域网里面，但是可以通过路由器端口映射的方法允许外网进行访问。

树莓派是小型化且在市场上最为廉价的通用计算机。最新版树莓派配有 1.4GHz, 64 位，四核 ARM Cortex-A53 的博通 BCM2837B0 SoC 处理器，1G RAM (动态访问内存)，4 个 USB 接口，HDMI，声卡，8P8C 以太网接口，支持视频音频输出，支持 SD 卡，U 盘，硬盘，键盘和鼠标扩展，支持 WiFi 和蓝牙功能但是售价仅为 35 美元^[12]。就像其他任何一台运行 Linux 系统的台式计算机或者便携式计算机那样，利用树莓派可以搭建起

来一个云盘服务系统，再加上树莓派可以外接 USB 硬盘（如图1），这意味这我们可以外接硬盘来获取近乎无限的存储空间。



图 1: 树莓派单片机 2T 硬盘

1.1 研究背景

(1) 终端存储空间不足 对普通的个人消费者用户来说，移动手机还有个人计算机是两种比较常见的终端设备。用户在平时的工作，学习，娱乐中，会积累下来许许多多的文件，视频，音乐，图片，文本文档，游戏，还有下载的各种资源，这些文件通常占据了我们绝大部分的存储空间，我们电脑上的存储空间，手机上的存储空间，往往不够用，这时候大家就会想到，将这些文件备份到云端硬盘中，为本地节省下来存储空间^[3, 13, 14]。

现在市面上主流的手机存储容量一般有 16G, 32G, 64G, 128G, 256G, 然而对于手机用户来说，这些容量往往是不够用的。现在的手机应用占用的存储空间非常容易能达到 300MB 以上，QQ 微信的空间会更多一点。QQ 微信微信照片内容分享，再加上现如今的如今比较流行的短视频应用，摄像录影对用户来说可以说是绝对的刚需了。这时候一个普通的用户平时的照片视频存下来的话，没几天消耗的存储空间都能达到上百兆，一张照片有 3 到 5 兆的大小，用手机拍摄的没有经过后期的视频一分钟都有 150M 左右，很快的手机终端的存储空间被消耗殆尽。当然手机的厂家都会提供自身的云服务，但是在超过免费的存储容量之后，手机厂商就回收取非常昂贵的云存储费用。

相比使用闪存技术来存储数据的手机来说，个人计算机一般采用 128GB，或者 256GB 的 SSD 固态硬盘作系统盘，外加 500GB，或者 1TB 的机械硬盘来做数据存储盘。因此对电脑相对手机来说，存储设备的价格相比手机还是要低一些的，下面是一些固态和机械硬盘的价格图标，机械硬盘的价格还是可以接受的，但是在终端设备存储数据有一个很大的弊端就是数据的容灾不能得到保证，因为终端损坏而导致数据丢失用户后悔不及的事情屡屡发生。因此本项目数据和终端的分离就显得非常有现实意义了。

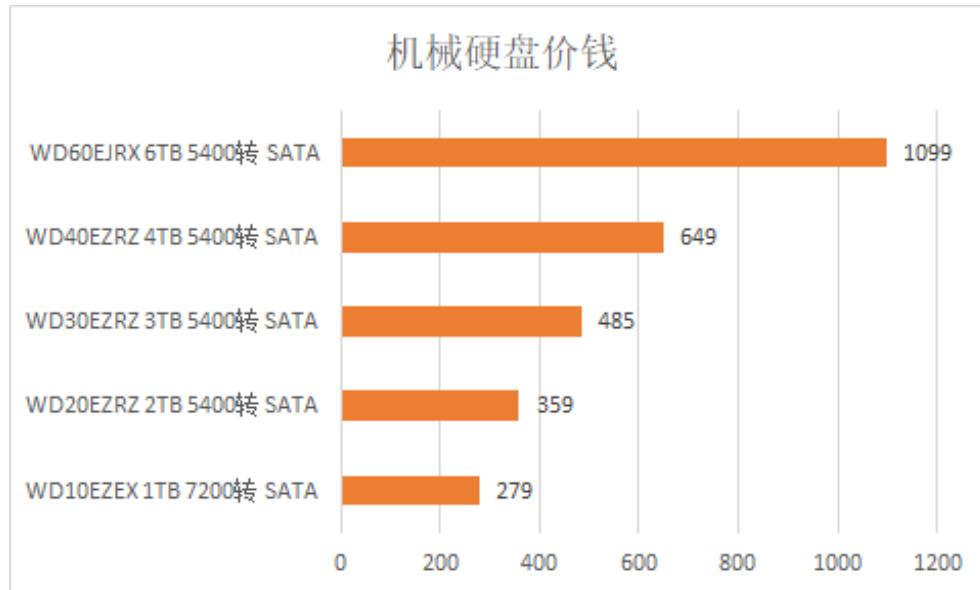


图 2: 西部数据硬盘价钱^[15]

(2) 云盘服务价格昂贵 云存储是一项刚需，现在各手机厂商都会有自家的云存储，像苹果的 iCloud, 小米的 Mi Cloud 等，但是价格并不便宜。苹果的 iCloud 的一年 2TB 的存储空间的价格为 810 元^[16]，小米 10G 就要收费 60 元，现在市场上比较流行的是百度云，百度一开始有 3TB 的免费的存储空间，但是如果你额外要存储更多的数据的话，可以向他购买一年 2TB 价格为 249 元的存储空间套餐，但是百度云对非会员用户来说下载速度有很大的限制，现在百度云将非百度云会员的下载速度限制在 500KB/S 以下。

国外比较流行的云存储服务，有谷歌的 Google Drive，微软的 One Drive，亚马逊的 Amazon Cloud Driver，但是国外的云存储服务有两个很明显的缺点，一个是价格，比百度云还要昂贵，而且因为特殊的国情这些国外的云存储厂家都没有在国内开服务器，在网络的稳定性和速度上有很大的折扣^[17-20]。相比于各厂商提供的方案，本系统可以提供一种更为经济的云盘解决方案，就是通过树莓派（英语：Raspberry Pi）和家庭宽带部署云盘服务。闲鱼上 1 块 2TB 的二手机械硬盘，大概需要 250 块钱左右，我们知道一块机械硬盘的寿命可以维持在 4 到 5 年左右，平均下来每年只需要花 90 多块钱就可以享

受到 2TB 的存储空间，一个私人的云存储服务，可以花更少的价钱而得到更大的存储空间。

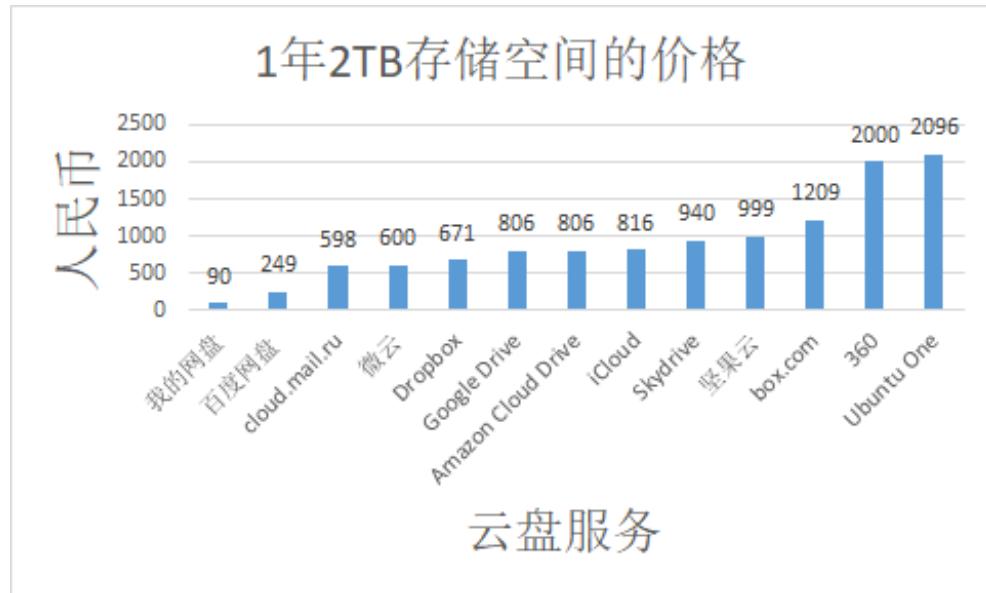


图 3: 市场上主流的云盘服务价钱^[16-21]

当然私人的云盘服务有很大的缺点就是，在文件共享，异地访问，并发访问，相比于成熟的商业公司提供的云存储服务有很大的缺陷。本项目是利用上海电信提供的 20 兆家庭宽带，服务器上载速度明显限制在 500KB/s 以下。在本项目的实验中，在北京异地访问寝室树莓派上的视频文件只能够保证 20KB/s 的速度，效果不能令人满意。但是局域网环境下，本系统基本可以做到 1.25MB/s 的上载下行速度。

(3) 家庭宽带和磁盘资源没有得到充分利用 家庭宽带和硬盘资源很难得到充分的利用。因为学习和工作的原因，用户不在家时间占了大部分的时间，宽带资源往往得不到充分的利用。同时因为硬盘技术的发展，旧电脑的机械硬盘被固态硬盘取代，这些淘汰下来的机械硬盘也得不到充分利用。本系统的目的之一就是开发一种在线存储系统，将这些淘汰的资源充分利用起来。

关于使用旧的硬盘有硬盘损坏数据丢失的风险。机械硬盘的理论寿命大约三万个小时，企业级硬盘、监控用硬盘，如果始终保持高速运转工作，三年的话就会损坏，消费者级的硬盘正常使用 6 到 7 年是没有问题的，不过硬盘的寿命的预测大多基于生产数据的结果统计，相关的理论分析还是很缺乏，也没有相应的研究成果公布^[22]。一般机械硬盘的寿命和它的读写次数有关，如果频繁读写的话，机械硬盘的寿命会大大的降低，但是本项目的磁盘独显磁盘读写的场景并不多，而且本项目还做了相对应的增量备份，异地备份，在一定程度上保证了数据安全。

(4) 文件增量备份、异地备份 文件备份的重要性不言而喻，一旦发生灾难或错误操作的时候，我们需要一种手段能够有效的恢复系统的数据。在文件备份手段方面，有三个主要的形式，完全备份、增量备份和差异备份，如何在确保数据安全的基础上能够尽量减少存储空间的消耗，这是本系统要解决的难题。另外，如果要确保数据的容灾，需要做到异地备份，本系统原始数据是存放在家里或者寝室中的机械硬盘中，发生数据损坏的概率很小，但是数据的重要性是不言而喻的，所以需要做到一点在异地备份的方案中，有两个可选择的方案，一种是成本比较低的，就是在寝室或者家中各有一个服务器进行备份，另外就是使用亚马逊的 aws 的 s3 存储服务，使用云存储的安全性更高，但相应的备份速度更慢而且成本更高^[23]。有一种方案是二者的结合，在家中和寝室的服务中进行备份的同时，有重要标签的文件使用商业公司提供的云存储服务，这是一个安全和成本兼顾的方案了。

1.2 本文贡献

1.2.1 云盘系统实现

本文实现了一个提供文件备份、共享、存储、访问、搜索、管理等功能，满足多终端使用的且部署在单片机树莓派上在线云盘存储系统。从图4可以看出树莓派体积非常小，仅有大概 4 英寸的大小，这意味着它的能耗非常低。实验数据表明树莓派的平均功耗只有 2W/h，意味着如果全年运行树莓派仅仅需要消耗 18 度电，但是它的性能完全能够满足文件的上传备份访问、少量并发等基本功能，相比于个人计算机或者是专业的商用计算机来说，树莓派的功耗可以说是几乎忽略不计²⁶。本系统通过路由器的端口映射，提供外网访问的功能，用户可以通过安卓客户端或者 web 来访问存储在机械硬盘上的文件。经过测试，局域网内的文件上传能达到 19.7Mbps 的上载速度，20.4Mbps 的下载速度。本实验的路由器网卡速率为 500Mbps，基本上满足局域网的文件访问功能。经过测试，北京访问基于上海电信网络环境的树莓派上的文件可以达到平均 25Kbps 的速率，非常遗憾这个速率不能满足异地文件访问的要求。当然网速取决于运营商提供的网络套餐，没有必要在代码上进行优化，况且本系统的使用场景大多是在寝室局域网和校园网、城域网内，所以像文件的备份和远程访问都是没问题的。除了性能上的可用性，本系统还实现了 Web 客户端（图5、安卓客户端（图6），用户所需要的基本的云盘功能都有实现，所以本系统具备功能上的可用性。



图 4: 系统总览

1.2.2 Web 客户端实现

本系统 Web 端基于 jquery-ui 和 bootstrap 实现良好的用户界面，用户可以对文件进行增删改查的操作，也可以分享文件，添加标签，收藏文件，上传下载文件，基于 html5 视频在线播放，还有图片预览等功能。

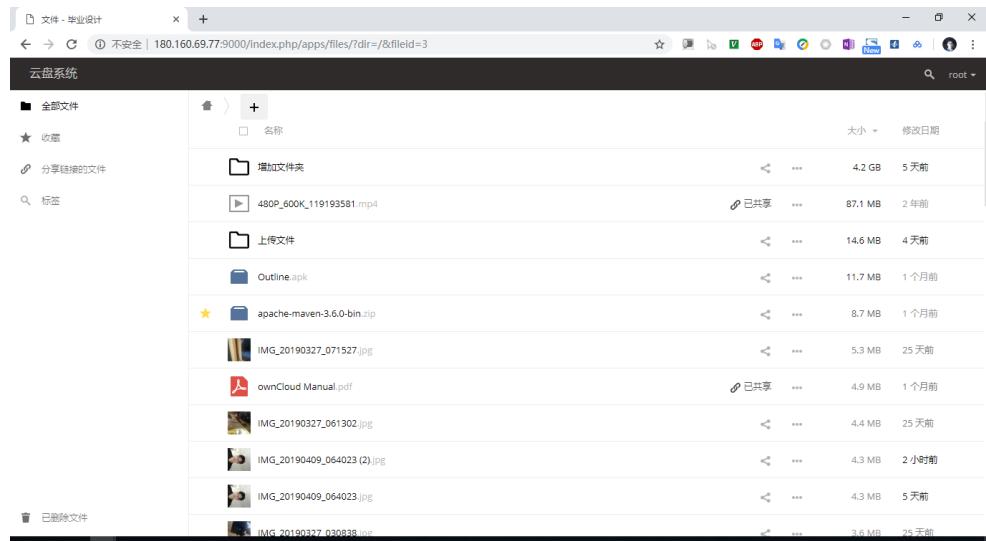


图 5: web 界面

1.2.3 安卓客户端实现

本系统安卓端基于 MaterialDrawer UI 框架实现良好的用户界面，基于 Volley 网络框架实现文件的上传下载，手机图片视频文件自动上传。



图 6: 安卓界面

1.2.4 安卓端拍摄图片、视频自动上传

用户可以选在开启或者关闭安卓端的文件自动备份，自动备份的原理是在安卓后台运行一个 Service 进程对手机/storage/emulated/0 下的目录如图7进行监听，当有新文件时会自动读取新文件，通过上传组件与服务器进行交互。因为这一个过程需要大量的后台资源，手机端的 app 是默认关闭的。当然为了提升用户体验，也可以设定为在特定时间进行文字备份，比如凌晨时间。

```
generic_x86:/storage/emulated/0 $ ls -l
total 80
drwxrwx--x 2 root sdcard_rw 4096 2019-03-08 13:02 Alarms
drwxrwx--x 3 root sdcard_rw 4096 2019-03-08 13:02 Android
drwxrwx--x 5 root sdcard_rw 4096 2019-03-11 21:36 DCIM
drwxrwx--x 2 root sdcard_rw 4096 2019-03-11 21:35 Download
drwxrwx--x 2 root sdcard_rw 4096 2019-03-08 13:02 Movies
drwxrwx--x 2 root sdcard_rw 4096 2019-03-08 13:02 Music
drwxrwx--x 2 root sdcard_rw 4096 2019-03-08 13:02 Notifications
drwxrwx--x 2 root sdcard_rw 4096 2019-03-08 13:02 Pictures
drwxrwx--x 2 root sdcard_rw 4096 2019-03-08 13:02 Podcasts
drwxrwx--x 2 root sdcard_rw 4096 2019-03-08 13:02 Ringtones
generic_x86:/storage/emulated/0 $
```

图 7: 需要监听的外部存储的目录

1.2.5 文件增量备份、异地备份

系统需要至少两台机器来进行文件备份，机器间需要保持数据的同步，如果数据不一致，需要快速的定位到不一致的节点。本系统实现了在每台机器上针对每个目录数据的散列值维护一棵 Merkle 树，这样在两台机器间进行数据比对时，从 Merkle 树的 root 节点开始进行比对，如果 root 节点一样，则表示这两台机器间的目录数据是一致的，不再需要任何处理；如果不一样，则遍历 Merkle 树，定位到不一致的节点也非常快速。相比于无脑地对文件进行完全备份，使用文件的哈希值作为文件的身份标识来构建 Merkle 树的做法提升了备份的速度和存储空间。

1.2.6 文件收藏、标签、搜索

用户可以为文件打上标签，或者收藏，系统会提供额外的分享，收藏，标签，垃圾箱目录给用户访问，用户不用辛苦寻找标记过的文件。同时系统还提供文件名模糊搜索（文件限定于在本目录搜索，如果采用和 Windows 的 Explorer 的深度全局查找，系统开销会非常大，而且效果不好，一般用户都会记得特定文件的存放目录，实现文件全局查找实在没有工程上的必要），标签查找。查找结果会动态的在前端页面展示出来，用户体验非常良好。

1.3 研究思路

本文从需求分析、概念结构设计、逻辑结构设计、编程实现、实验验证、测试运行几个方面展开研究。

(1) **需求分析** 需求分析阶段要对整个云盘系统有一个清晰明确的认识，将各个需求分割成可实现的各个小模块，明确系统的使用场景和面向的使用人群，指定开发的标准规范，况且系统的设计和实现目标，正对具体业务进行分析和差分，做好业务逻辑说明，明确系统角色，同时要明确哪些需求是功能性需求，哪些是非功能性需求，这样才能明确重点，提高开发的效率。

(2) **概念结构设计、逻辑结构设计** 主要采用 E-R 模型进行设计，包括画 E-R 图，然后通过将 E-R 图转换成表，实现从 E-R 模型到关系模型的转换，再然后为所设计的数据选择合适的存储结构和存取路径。

(3) 编程实现 选定好技术框架后就可以马上动手实现系统了。本系统的技术框架大概有 PHP,Apache,MySql,jquery,jquery-ui,bootstrap,Android,Volley, MaterialDrawer UI 等。

(4) 实验验证、测试运行 实现主要比较在不通网络环境（局域网，广域网），异地访问，本地访问， WiFi 环境，校园网环境、基站流量下对云盘系统的上传下载的速度和稳定性等基础功能进行测试。功能性测试方面要同时完成网页端和安卓端的单元测试、集成测试、系统测试、性能测试，安全测试、兼容性、接口测试、极限测试等。性能测试方面要完成对预期性能指标、核心模块并发、组合模块并发、大数据量、疲劳强度、网络性能等多个场景的测试。并通过实验得到实验数据来完成对系统的可用性和安全性的验证。

1.4 论文组织

本文一共分为六个章节，其结构安排如下：

第一章为绪论部分。讲述了本文的研究背景，阐述了本文的研究内容和贡献，并对研究结果做了简单的展示。

第二章为系统需求分析部分。结合文件管理系统实际，分别对系统的总体性需求、功能性需求、非功能性需求、系统技术可行性等几个方面进行介绍和分析。

第三章为系统功能设计部分。在需求分析的基础上对云盘系统进行总体设计，首先介绍了系统设计的目标和原则，在基于以上原则的基础上分别对系统的服务器架构、使用的一些技术框架，和数据库表格做了详细的介绍和分析。

第四章为系统功能实现部分。在完成功能分析和系统设计之后，此部分详细介绍展示了系统实现的结果，ui 界面，系统操作步骤，和一些关键功能的算法实现。

第五章为系统测试与分析部分。此部分结合本系统的实际对一些常见的测试方法进行粗略的介绍，同时详细介绍了本系统测试时所需要的测试环境和测试工具，然后本文设计了一些关键的测试用例，并对测试结果进行分析和总结，最后本文用 Linux 工具 dstat 对服务器的性能进行粗略的分析。

第六章为总结与展望部分。对本文的主要工作以及创新点进行总结，对未来研究进行展望。

2. 需求分析

2.1 系统总体需求分析

个人计算机和移动手机的存储不足是非常常见的现象，人们迫切地希望能有额外的空间对个人数据进行备份同时还可以方便地访问云端文件。本系统的目的就是为了解决上诉用户的痛点，同时将闲置的网络和硬盘资源利用起来，起到积极的环保和省钱效果。整个系统具有下面的优点

- (1) **完善的功能** 本系统的个人网盘系统具有文件上传下载、公开分享、用户登录注册、视频在线播放、web 端安卓端数据同步、手机文件自动备份、文件标签、文件搜索、无线添加硬盘存储空间、文件操作符合用户平时在 windows、安卓平台习惯的文件操作方式，用户可以没有学习成本就可以使用本系统，本系统具有以上诸多优点。
- (2) **先进的文件管理方法** 本系统具有明确的管理方法，通过计算机网络技术为用户提供近乎无限的存储空间，解决用户数据丢失的担心，真正做到方便好用的云盘系统。
- (3) **成熟的技术** 本系统基于主流的系统架构 LAMP(Linux、Apache、Mysql、php) 来开发，系统具有性能稳定、可扩展性高、安全可用等优点。
- (4) **操作方便快捷** 本系统提供 web 和安卓两种客户端访问方式，界面美观，操作逻辑符合用户习惯，用户无需特别学习，凭借之前在 windows 和安卓平台积累的习惯就可以很快学会使用本系统。经过测试，在局域网的网络环境下，用户上传下载文件，在线视频都可以以接近网卡的速度进行服务器访问，在广域网的网络环境下，本系统仍然可以提供基本的访问条件。
- (5) **安全兼容性高** 经过测试，本系统的 web 端在主流的浏览器 Chrome、Edge、IE11、QQ 浏览器、360 安全浏览器、360 极速浏览器、百度浏览器、opera 浏览器、手机端的 uc 浏览器、小米系统浏览器等浏览器都可以正常访问。同时系统的设计具有坚强的安全保密功能对用户个人信息进行保护，并提供良好的接口，使得文件可以方便的导入导出。

2.2 功能性需求分析

在本章节中会介绍本系统所涉及到的功能需求，同时为了方便描述需求，引入了三种用户角色，一种是 Web 用户、一种是安卓用户、另一种是非系统用户。非系统用户针对的是所有互联网用户，因为本系统涉及文件的分享，并且提供外网访问本地文件的方式，因此，作为非本系统用户还是可以访问到系统用户分享后的文件的。

2.2.1 用户控制需求

在需求分析开始之初设计了两种注册方式和 3 种登录方式如下
系统需要支持 2 种注册方式：

- 1) 账号；
- 2) 手机号，用户输入相应的注册信息完成注册。

系统需要支持 3 种登录方式：

- 1) 通过账号登录；
- 2) 使用手机号直接登录；
- 3) 通过第三方平台登录，例如微信、QQ 等。对于使用手机号登录。

但是在后期的开发过程中，发现注册和登录的功能并不是被系统所关键需要的，因为本系统的定位是局域网内家庭或者寝室成员之间交互，所以完全可以以纯后端注册的方式来节省开发时间，因此在基于时间紧迫只实现了账号口令登录的功能。

如下用户用例图，web 用户和安卓用户都可以进行登录、修改信息等操作，非系统用户不被赋予此项权限。

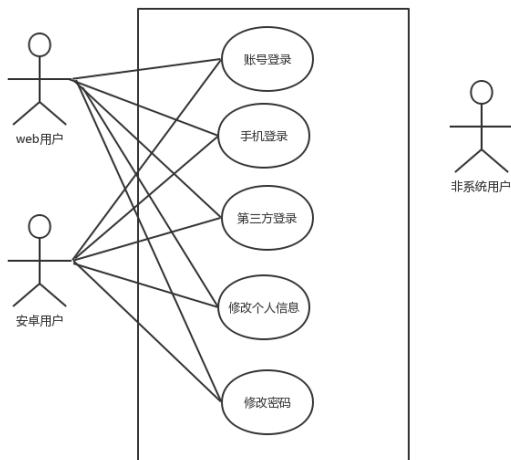


图 8: 用户登录管理用例图

2.2.2 文件/目录操作需求

本系统提供云存储服务，自然有创建文件/目录、读写文件、删除文件/目录、访问文件/目录属性等。

如下用户用例图，

web 用户存在文件上传、文件下载、或许文件详细信息、重命名文件、删除文件、查找文件、分享文件等需求。

安卓用户存在文件上传、文件下载、或许文件详细信息、重命名文件、删除文件、查找文件、分享文件等需求。

非系统用户可以访问下载被分享的文件，但是没有其他操作权限。

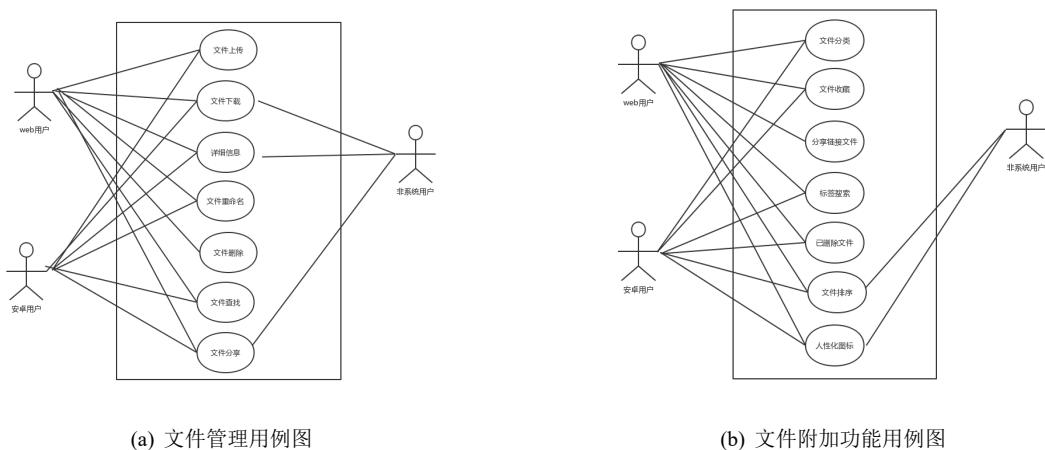


图 9: 文件操作用例图

2.2.3 文件/目录分享需求

文件或目录分享是本系统一个非常重要的需求，市场上主流的云盘服务都有提供文件分享功能，同时为了系统安全考虑，在实现系统的时候需要做一些安全性措施。

如下用户用例图，

web 用户存在设置分享链接名称、设置分享密码、设置文件过期时间、修改分享信息、社交分享、访问下载分享文件等需求。

安卓用户存在设置分享链接名称、设置分享密码、设置文件过期时间、修改分享信息、社交分享、访问下载分享文件等需求。

非系统用户可以访问下载被分享的文件，但是没有其他操作权限。

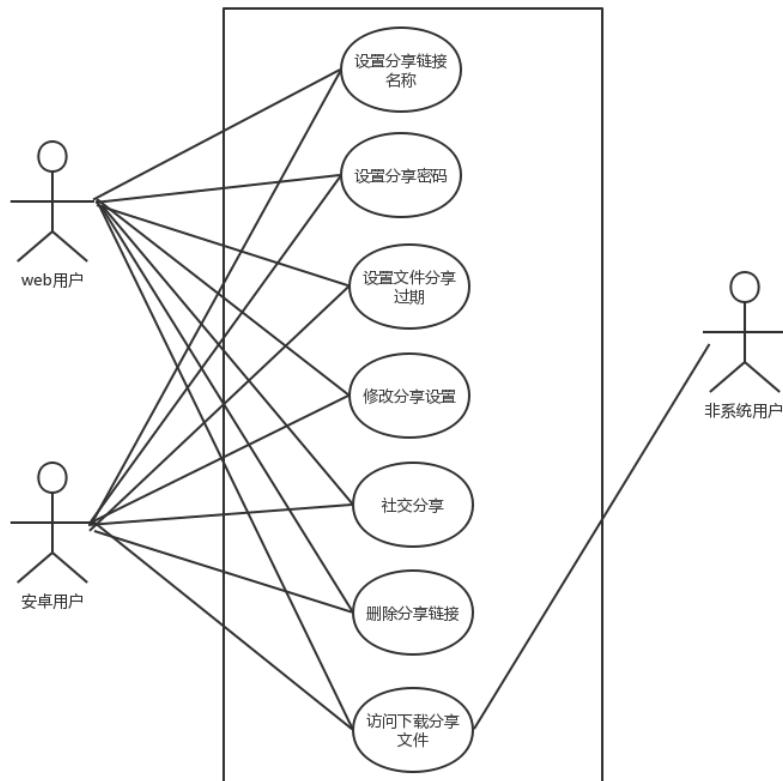


图 10: 文件分享用例图

2.2.4 视频在线播放需求

视频和游戏、社交并称当今互联网三大主流需求，因此在线播放视频是本系统必须要实现的需求了。本系统采用 Apache 作为服务器后端，对静态资源的访问有很好的支持，利用 html5 提供的 video 元素就可以很方便地实现流媒体播放的功能，同时 Apache 对安卓端的 VideoView 也有很好的支持。

如下用户用例图，

web 用户存在视频播放，图片预览、图片视频下载等需求。

安卓用户存在视频播放，图片预览、图片视频下载、图片自动备份、视频自动备份、 WiFi 环境上传文件、备份后删除原文件节省终端空间等需求。

非系统用户可以访问播放、预览、下载被分享的图片视频文件，但是没有其他操作权限。

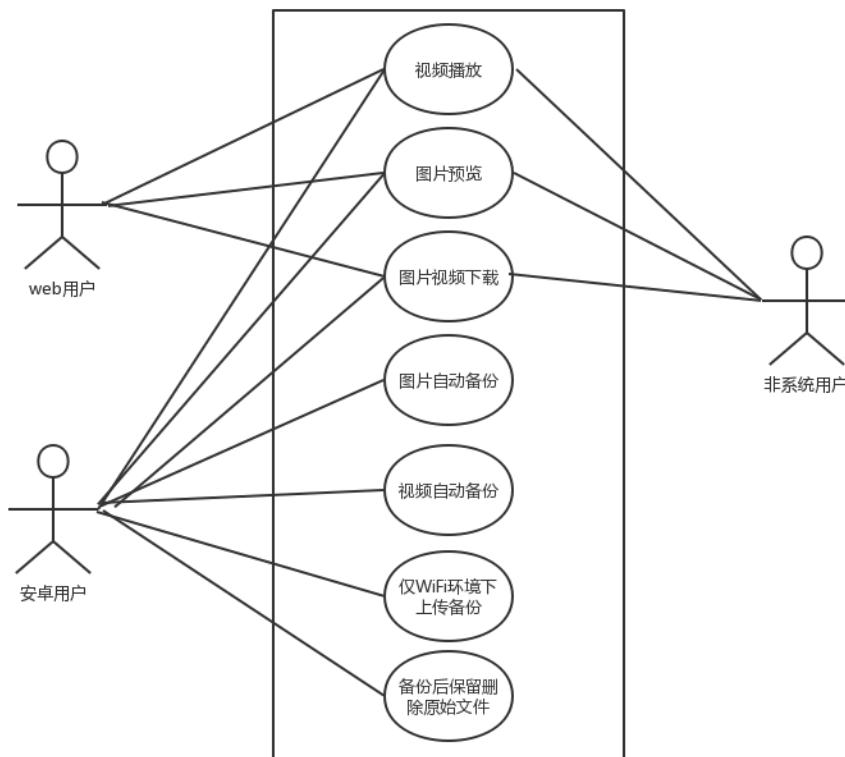


图 11: 视频图片文件用例图

2.3 非功能性需求分析

(1) **响应时间** 响应时间的定义是指一个客户端发送请求到服务器，然后接收服务器响应至结束的这一个过程中总共耗费的时间。响应时间包括三个部分，客户端呈现时间、网络传输时间和服务器响应时间。其中客户端的呈现时间是指客户端在接收服务器返回的数据后然后渲染成可视化页面所消耗的时间，然后服务器的响应时间是指服务器从接收到客户端发来的请求，然后在经过一些逻辑操作后（比如查询数据库，业务判断等）向客户端返回数据所消耗的时间。

本系统的使用场景一般位于局域网的环境下，响应时间需要做到基本处于 2 秒以下。同时如果处于外网的网络条件下，访问带宽也要达到中国电信所给予的 500KB/S 上载速度，响应时间也需要做到 5S 以内。同时本系统是云盘系统，提供文件上传下载功能，因此网络的稳定性和速度都是重要的考量。其实就树莓派和路由器来说，网卡速度最大 500Mb/s，因此在软件算法上提升速度没有太大的意义，后期测试过程中，大文件的上传下载速度基本能保持在 1.25MB/s 左右的速度。

(2) 并发用户数 并发的定义是多个用户在统一的时间点发送同一件事情或者操作的请求，这些操作一般都是相同的事务行为。比如用户在互联网抢票的时候，会频繁的刷新结果，导致服务器不堪重负。再比如在文件上传的业务中，不同的用户可能上传相同名称的文件并且在同一路径下，这时候就需要对该文件加锁才能保证后来的文件不会覆盖已有的文件了。同时，多个用户同时上传文件对孱弱的家庭宽带来说是不小的挑战，因此并发量达到 3 个并发用户就可以满足本系统一开始的设定了。

(3) 吞吐量 吞吐量的定义是指单位时间内系统接收并处理用户发起请求的数量，同时能够非常直观地体现服务器系统的性能承载力。对于具有交互式性质的应用系统来说，吞吐量可以体现服务器系统所承受的性能压力。在各种测试场景中，吞吐量一直都是一个重要指标，它可以在 database、middleware 和计算机硬件上得到直观的体现。系统性能瓶颈可以用吞吐量来协助分析，还可以在设计软件性能测试场景用到吞吐量指标来衡量软件性能是否达到了预计的设计目标，比如交互式应用系统中的 Connection Pool、数据库事务发生频率、次数等。

(4) 资源利用率 资源利用率是指对计算机系统硬件和网络资源的利用程度，例如服务器的 CPU 利用率、内存利用率、磁盘利用率、网络利用率、GPU 利用率等。同时在改善系统性能的时候可以用资源利用率作为分析判断的重要指标。

在本系统的性能需求章，需要满足下列的性能指标。

- (1) CPU 使用率上限不超过百分之八十五
- (2) 内存利用率上限不超过百分之八十五
- (3) 磁盘 I/O 交换率上限不超过百分之八十五
- (4) 网络带宽平均可保持在 1.25MB/s 以上

2.4 系统可行性分析

2.4.1 操作可行性分析

本系统的操作界面力求简洁明了，简化操作流程，使操作尽量符合用户的使用习惯和文件管理规范，用户无需特别学习就简单上手，因此本系统的操作可行性是绝对可以的。

2.4.2 技术可行性分型

LAMP (Linux、Apache、MySQL、PHP) 技术架构在世界范围内有非常广泛的应用，根据统计数据，世界上大约百分之七十 2 的网站都是基于 LAMP 技术框架搭建。LAMP 由 Linux 系统，Apache 容器，Mysql 数据库，PHP、Python 等动态语言组成一个非常轻量和灵活的技术框架，开发环境的搭建也非常方便，在 github 上有一键安装 LAMP 的脚本项目，受到世界上不少开发者的欢迎^[24]。同时因为这个技术框架中所有组成产品均是开源软件，很多互联网企业的商业应用都是采取这个架构，在开发成本和速度上有着明显的优势。J2EE 也是开源产品，但是 J2EE 架构源、轻量、快速开发这些方面和 LAMP 相比处于劣势。微软的 dot NET 架构在通用、跨平台、性能、价格等方面也是明显不如 LAMP 架构，因此 LAMP 无论是性能、质量还是价格都是开发本云盘系统的第 一选择^[25, 26]。

本系统 LAMP (Linux, Apache, Mysql, PHP) 架构模式中，Linux 基于 Debian 版本，具有稳定性强，升级方便，软件包丰富，性能优越，开发简单等优点。同时采用 Apache 作为服务器，是因为 Apache 对静态资源的访问非常友好，避免了开发对静态资源的手动处理，大大节省了开发时间。服务器规定了统一的 API 接口，无论是 web 客户端，还是手机 APP 客户端都可以采用统一的接口访问到服务器的资源，大大提升了开发效率。同时数据库采用了开源的 Mysql 数据库，树莓派上 Mysql 版本是 mariadb，相比于 Oracle, MySQL 具有开源免费，轻量易用，开发方便的优点，是互联网最流行的数据解决方案。在本云盘系统中，所有用户信息、文件路径信息、分享文件信息、系统配置信息和其他类型的信息都存储于数据库中，通过 SQL 语言可以很容易地查询这些信息。PHP 是开源的动态语言，它既可以用来做后台业务逻辑的开发，也可以嵌入到 HTML 中实现前端效果，也可以访问 MySQL 数据库中的数据和 Linux 提供的一些特性的动态内容。因此 PHP 非常适合 web 开发和为 Android 数据访问提供接口。同时因为服务器是部署在中国电信给予的局域网内，所以如果需要外网访问，就需要进行端口映射，端口映射的时候要避免将外网端口设置成常见的 80 端口，因为中国电信会为了网路安全，对特定端口不允许开放。同时系统后台会开放一个备份进程，定时对文件进行备份。

3. 系统设计

3.1 系统设计目标和原则

本系统旨在设计和开发一个适合少量并发，大部分局域网场景下，提供文件的上传、下载、分享、搜索、备份、视频播放等功能，并且使这些功能在实现后易于上手使用，运行稳定并且具有良好的用户体验和可靠的安全性。因此在设计系统中过程中必须遵循以下原则：

符合开发规范：遵循系统开发各项相关标准和体系。

选用成熟技术方案：为了保证开发效率和开发质量，成熟的技术方案可以保证系统的稳定性和低学习成本，因为网上的资料会比较多。

完善的功能：在需求分析阶段提出的功能尽量完成，为用户提供可行的解决方案。

可扩展性强：系统开发时长期的过程，需求也是不断发生变化，因此设计的时候尽量采用设计模式，预留接口，二次开发才能高效进行。

维护简单：系统需要长期稳定运行，而长期稳定的运行离不开日常的维护。

界面简洁，功能易用。

3.2 服务器设计

本云盘系统的服务器设计采用经典的 LAMP 架构，处理一次客户端动态资源请求的过程如图12 所示，

(1) 服务器处理流程

- a) Client 发起 http 请求
- b) Httpd Server Apache 接收请求并交给 CGI 进程处理
- c) 通过 CGI 接口访问 PHP 的的应用程序
- d) PHP 应用程序调用 PHP 解释器执行 PHP 代码
- e) PHP 程序通过 Mysql 驱动访问数据库数据
- f) PHP 对数据进行封装，最后给客户返回相应

故在 LAMP 的环境机构中，apache、mariadb 和 php 的主要功能分别如下。

(2) apache 主要实现如下功能：

- a) 处理 http 的请求、构建响应报文等自身服务；

b) 配置让 Apache 支持 PHP 程序的响应，修改 apache conf 目录下的 httpd.conf 配置文件，引入 PHP 模块；

c) 配置 Apache 具体处理 php 程序的方法，在本系统中统一使用路由表来出来方法和路径映射；

(3) mariadb 主要实现如下功能：

- a) 提供 PHP 程序对 Mysql 数据库数据的写入；
- b) 提供 PHP 程序对 Mysql 数据库数据的读取；
- c) 通常情况下从性能的角度考虑，尽量实现数据库的读写分离，但是本系统地位是个人的云盘系统，因此并发场景不多，所以没有实现读写分离

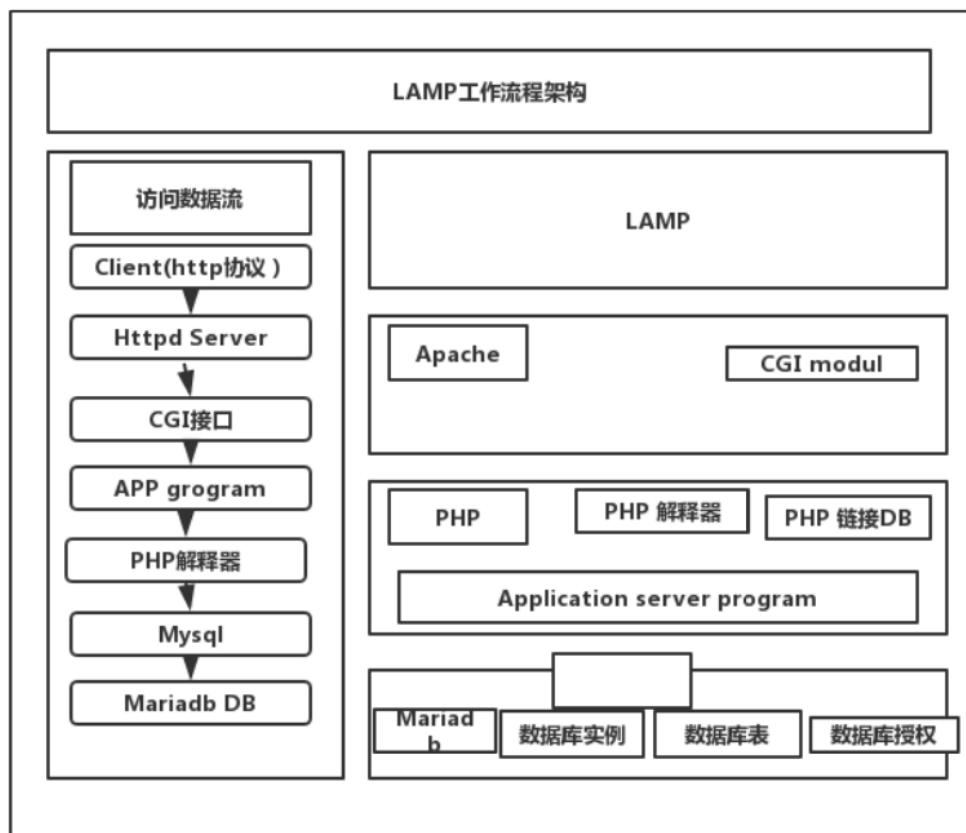


图 12: LAMP 架构图

(4) php 主要实现如下功能：

- a) 实现 Apache 的访问接口 CGI，一般用户请求都交给 CGI 处理，相当于 Tomcat 中的 servlet 方法。对于 Apache 对静态资源的访问可以不经过 CGI 接口；
- b) 提供 PHP 程序的解释器；
- c) 提供 Mysql 数据库的访问驱动，驱动是为了将数据库的磁盘比特转化成 PHP 可

以识别的数据结构。

具体的服务器设计如图4所示。经过路由器端口映射，客户端可以在外网或者局域网的网络环境下访问服务器，在服务端，系统实现了动静资源的分离，如果是直接访问文件，直接经过 apache 的路径就可以不需要 php 来处理请求了，直接返回静态文件资源。同时系统会添加新进程，进行文件的增量备份和异地备份。

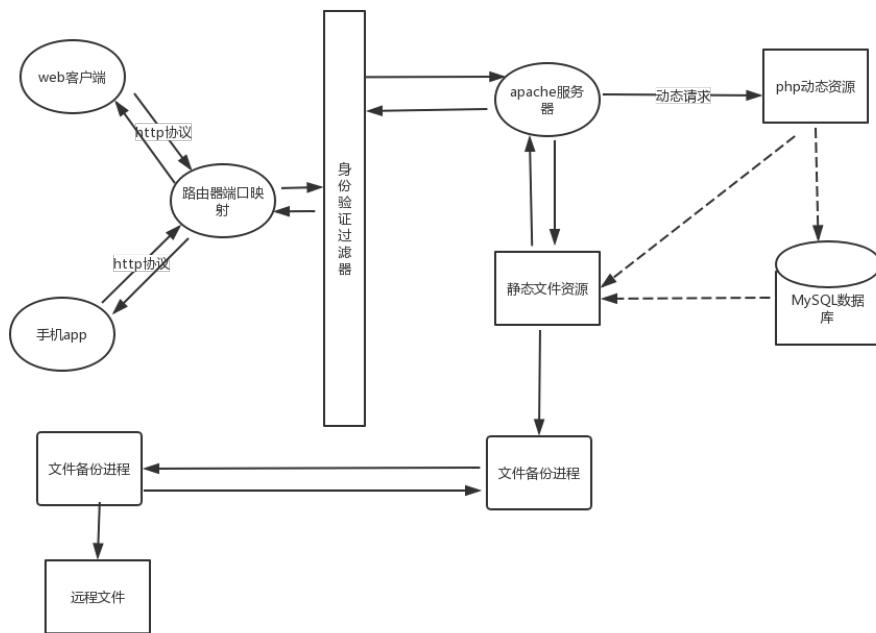


图 13: 系统总架构图

3.3 数据库设计

根据系统的各个功能模块的划分和各个系统角色的从属关系，设计了一下的数据库系统，第一部分是从全局的视角展示各个系统角色之间的关系，接下来再详细地介绍各个表字段的含义。

本文的系统数据库表格设计主要分为 3 个模块，分为用户模块、文件模块、系统配置模块。用户模块和文件模块有依赖关系，系统配置模块需要规定服务器系统的硬件配置、允许访问的外网 ip、内网 ip、文件根路径、存储空间大小等信息。用户模块包括 OcAccount、OcMounts、OcUsers 3 张表，文件模块包括 OcMinetypes、OcShare、OcFilecache、OcFileTrash、OcSystemtagObjectMapping、OcSystemtag 6 张表，配置模块包括 OcAppconfig、OcAppconfigKey、OcAuthtoken、OcAuthtokenWithBLOBS 4 张表。

在概念设计的基础上，可以在数据库中完成实际的建表，同时建立各个表互相之间

的外键联系。在数据库的设计过程中，需要遵循一定的设计规范，以减小数据的冗余，提高数据访问效率。本系统选用 MySql 5.7 版本的数据库进行开发，需要建立的表包含系统用户、文件类型、分享文件、垃圾文件、基本文件、文件标签、标签文件关系映射。遵循统一化驼峰法命名原则，每个表的命名都以 Oc(Own Cloud) 开头，最后部分为表名称的英文。云盘系统的核心功能的表物理结构如表1 2 3 4 5 6 7所示。

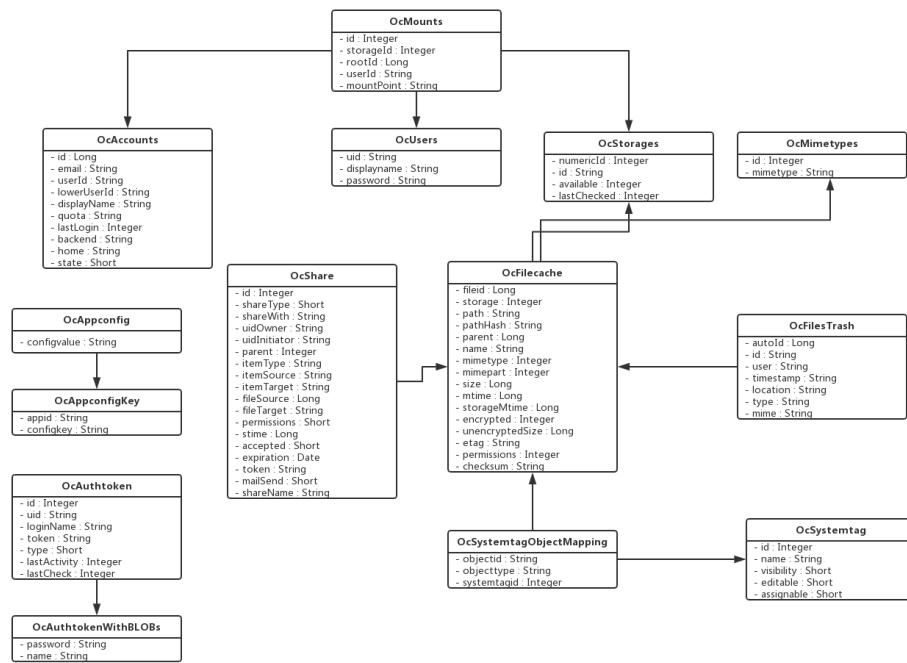


图 14: 数据库总类图

(1) 用户基本信息表 用户基本信息：包括用户 id、用户名、口令、邮箱、存储外键、上次登录时间、家目录、用户状态。

表 1: OcUsers 用户表

字段名	数据类型	说明
uid	String	用户 id
displayname	String	用户名
password	String	口令（哈希过）
email	String	邮箱
storageId	Integer	存储外键
lastLogin	Integer	上次登录时间
home	StringA	家目录
state	Short	用户状态

(2) 文件类型基本信息表 文件类型基本信息：文件存在视频、图片、文件夹、二进制文件、可编辑文本文件、音频文件等类型

表 2: OcMinetypes 文件类型表

字段名	数据类型	说明
id	Integer	id
mimetype	String	文件类型

(3) 分享文件基本信息表 分享文件基本信息: id、用户 id、分享文件是文件夹类型还是文件类型、文件路径、是否允许共享、分享时间、文件身份标识、分享文件名、分享链接过期时间等

表 3: OcShare 分享文件表

字段名	数据类型	说明
id	Integer	id
uid	String	用户 id
itemType	String	文件夹或文件
fileTarget	String	文件路径
permissions	Short	是否允许共享
stime	Long	分享时间
token	String	文件身份
shareName	String	分享名字
expiration	Date	过期时间

(4) 垃圾文件信息表 垃圾文件信息: 包括 id、用户 id、时间戳、位置等信息。用户可把不用的文件放入垃圾箱, 系统默认保存 1 个月的时间, 在一个月时间内, 用户可恢复文件, 否则系统会在物理存储设备中删除文件。

表 4: OcFilesTrash 垃圾文件表

字段名	数据类型	说明
id	String	id
user	String	用户
timestamp	String	时间戳
location	String	位置

(5) 基本文件信息表 基本文件信息: 这个表是系统的基本文件表, 对文件的增删改查就是基于此表。主要包括文件 id, 存储, 路径, 路径哈希, 父目录, 文件名, 文件类型, 文件大小, 上传时间, 源文件创建时间, 是否压缩, 压缩后文件大小, 文件标签, 文件哈希值等信息。

(6) 标签表信息表 标签表信息: 用户可以为每个文件打上标签, 方便日后查找, 同时用户可以为单个文件打上多个标签。主要有标签 id, 标签名, 是否可见, 是否可编辑等。

表 5: OcFilecache 文件表

字段名	数据类型	说明
fileid	Long	文件 id
storage	Integer	存储
path	String	路径
pathHash	String	路径哈希
parent	Long	父目录
name	String	文件名
mimetype	Integer	文件类型
size	Long	文件大小
mtime	Long	上传时间
storageMtime	Long	源文件创建时间
encrypted	Integer	是否压缩
unencryptedSize	Long	压缩后文件大小
etag	String	文件标签
checksum	String	文件哈希值

表 6: OcSystemtag 标签表

字段名	数据类型	说明
id	Integer	标签 id
name	String	标签名
visibility	Short	是否可见
editable	Short	是否可编辑

(7) 标签文件中间表 因为标签和文件之间是多对多的关系，所以需要一个中间表完成对二者关系的映射。标签文件中间表基本信息：文件 id、标签 id

表 7: OcSystemtagObjectMapping 标签文件中间表

字段名	数据类型	说明
objectid	String	文件 id
systemtagid	Integer	标签 id

4. 系统实现

4.1 系统实现环境

树莓派、2T 西部数据机械硬盘、机械硬盘电源、500Mb/s 网卡速率 TP-Link 路由器

4.2 系统功能实现

4.2.1 用户登录

登录注册需要一个很长的验证过程，下面的流程图给出很了很详细的验证过程。登录注册的流程图是功能需求分析的时候开始设计，考虑了很多因素进去，包括了更重成功、失败结果如何进行处理。但是因为开发时间紧迫，目前本系统只实现了用户名，口令登录部分，剩下的部分在之后的完善计划中会有进一步的工作。

因为本系统 Web 端和安卓转都使用同一的接口进行登录，所以服务器会返回一个同一的 requesttoken 作为身份验证。传统的浏览器登录方案中，是浏览器发送的登录请求在服务器验证通过之后，服务器发回 SESSIONID 和重定向地址，浏览器会自动将 SESSIONID 地存入 Cookie 中作为下次请求的身份验证。服务器端会使用基于动态代理实现的面向切面编程的思想对所有请求进行拦截做身份验证，一般服务器会在 SESSION 中存有登录成功后的用户信息，服务器端的 SESSION 相当于一张 hash 表，根据客户端请求头中 cookie 值来判断 SESSION 中是否保存有用户信息，如果没有，那么就说明发送请求的客户端处于非登录状态，那么服务器就会发回 402 代码，如果存在用户信息，那么就会将客户端请求放行。

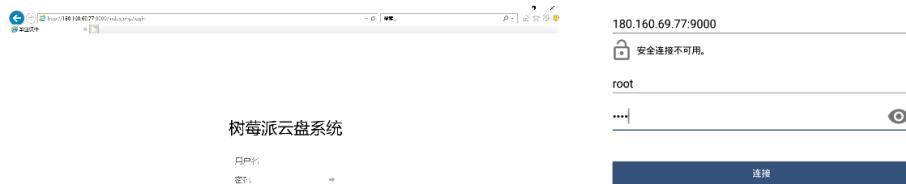


图 15: 安卓、网页登录界面

不同于传统的登录实现方案，本系统存在安卓客户端和 web 客户端，为了避免写

两套接口实现一样的功能从而带来开发效率的降低，服务器采用 restful 的思想统一 web 客户端和安卓客户端的接口。所以本系统的登录验证过程是这样的，客户端发送登录初始请求，服务端生成 requesttoken 并保存在 SESSION 中，并返回 requesttoken，在 web 端，浏览器会把 requesttoken 写入 Cookie 中。安卓端的网络层使用 Retrofit2 + RxJava + Okhttp3 框架，安卓端在接收到 requesttoken 后，会对 requesttoken 进行持久化处理，然后当每次使用 Retrofit 发送请求时，就会在 Retrofit 请求头中加入 requesttoken 的头部信息，从而达到身份验证的效果。

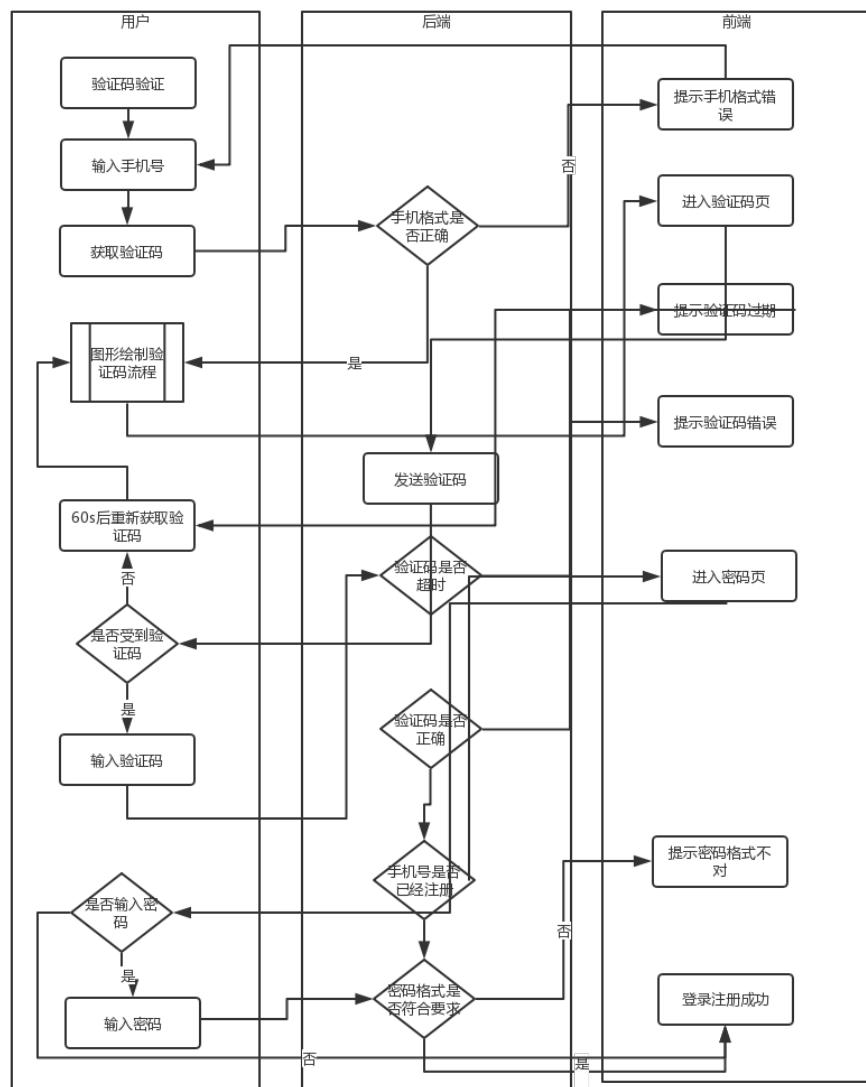


图 16: 登录注册流程

4.2.2 文件上传

要保证大文件上传下载的完整性，就需要断点续传的解决方案。

基本的思路是基于阻塞队列 BlockingQueue 使用生产者消费者模式，开启读文件线程将大文件切割成固定大小的 part，读取后存入阻塞队列 BlockinQueue 中，同时开启上传线程将 BlockingQueue 中 part 上传。BlockinQueue 为固定大小，在本系统中设置为 cpu 的核心数 4，如果 BlockingQueue 中的 part 数量大于 BlockingQueue 容量，则读文件线程陷入阻塞状态，如果 BlockingQueue 为空，则上传进程陷入阻塞状态，同时，如果读文件读完文件，会将 NULL 值加入 BlockingQueue 中，当上传线程读到 NULL 时，代表所有文件块已经全部上传，这时会通知服务器对文件进行合并^[27]。

下面给出了本系统使用多线程断点下载的流程图。

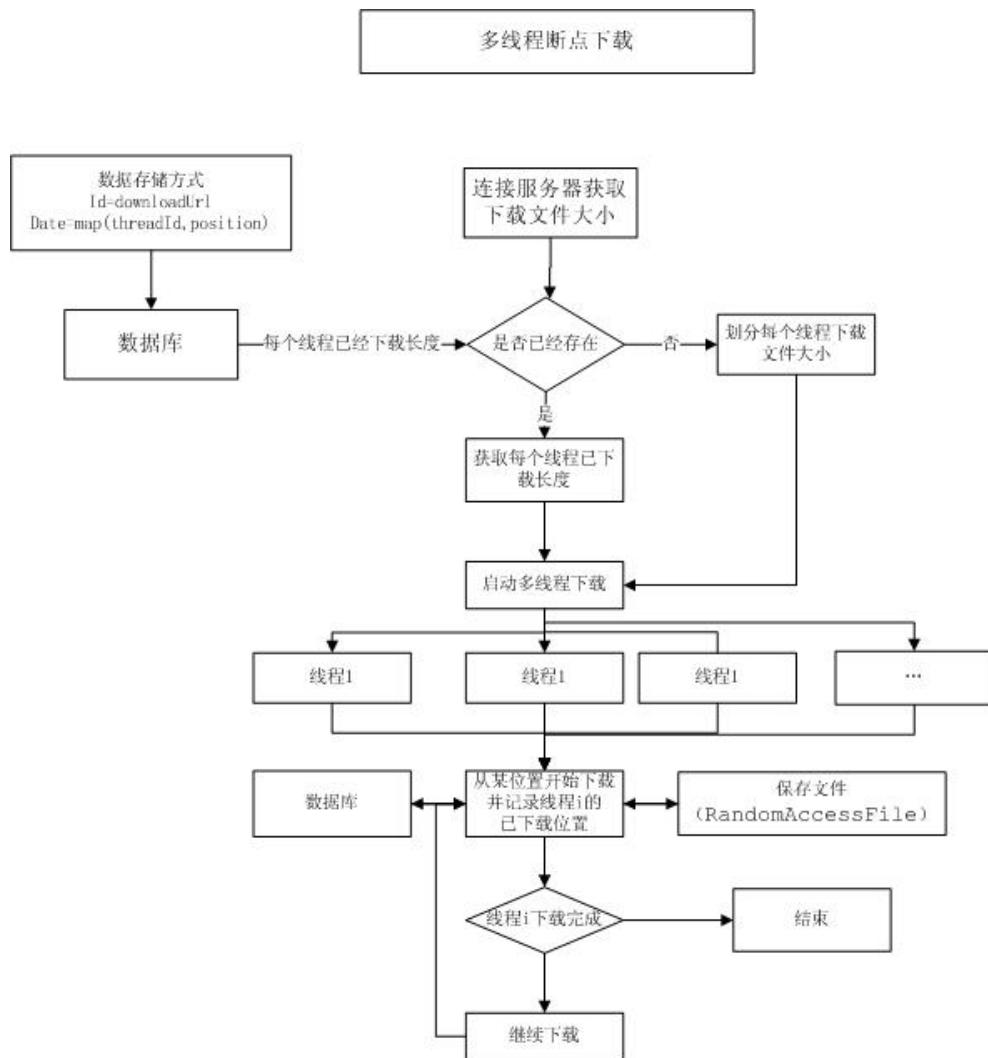


图 17: 断点续传下载实现

本系统提供 web 端和安卓进行文件上传。

安卓可以选择特定目录文件、及时拍照图片、最近修改图片进行上传。同时服务器会对上传文件加锁，如果上传的过程中用一个并发的线程上传相同路径下的相同文件名文件，那么改线程就会陷入等待轮询状态。如果发现上传的文件有相同文件名，那么就

会提示是否覆盖，如果用户选择不覆盖原有文件，系统会为新文件添加序列号来保证相同路径下文件名的唯一性。

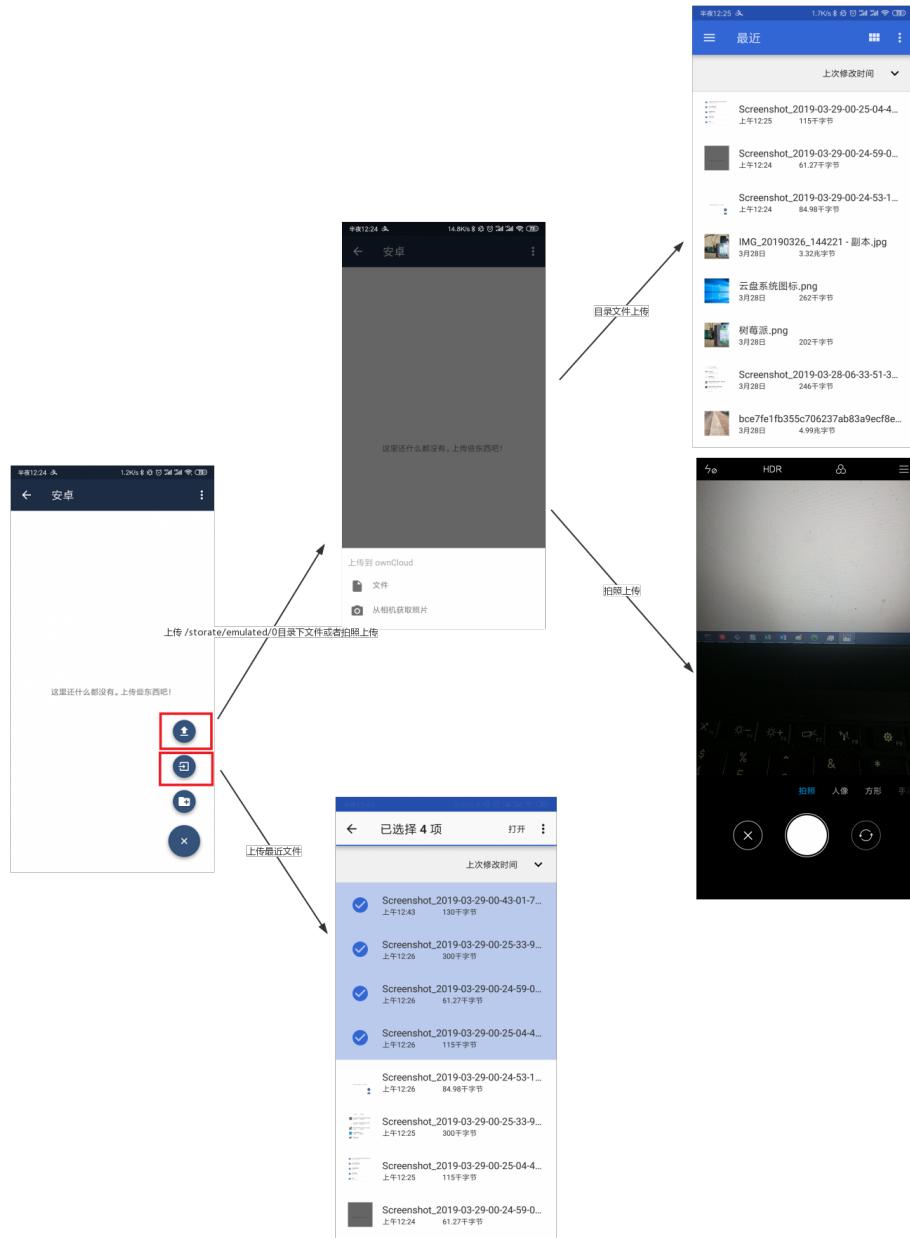


图 18: 安卓文件上传

4.2.3 文件下载

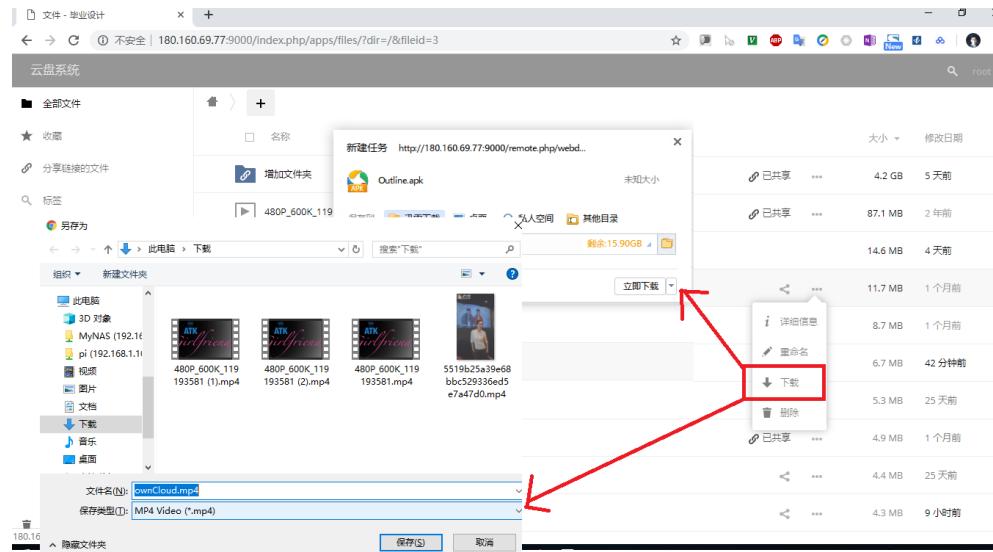


图 19: 网页文件下载

4.2.4 视频播放

web 端和 Android 端都支持视频播放功能，用户可以在线观看视频，web 端视频播放器基于 html5 的 videos 标签，可以直接访问在 apache 上的静态资源，省去了搭建流媒体服务器的麻烦。

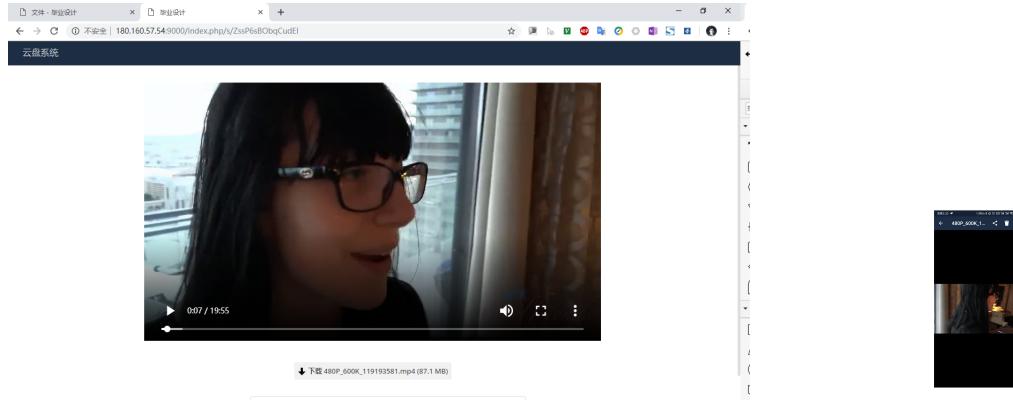


图 20: 视频播放

4.2.5 文件分享

用户点击创建公共链接，客户端发送请求给服务器，服务器首先判断公共链接是否已经创建，如果已经创建，则返回已创建链接，否则，利用 php 的 shuffle 随机函数根据文件 uuid 生成随机字符串存入数据库并返回客户端。客户单端根据返回信息对连接名称，

分享密码，过期时间进行初始化，用户可以修改链接名称、密码、过期日期等信息，发送回服务器，服务器将请求参数持久化到 mysql 数据库。

生成分享连接后，用户可以手动复制，通过社交软件或者邮件分享给互联网上的所有用户。安卓端有提供分享接口，通过实现系统接口可以调用包括 QQ、微信、短信、电子邮件或者其他第三方应用进行文件分享。

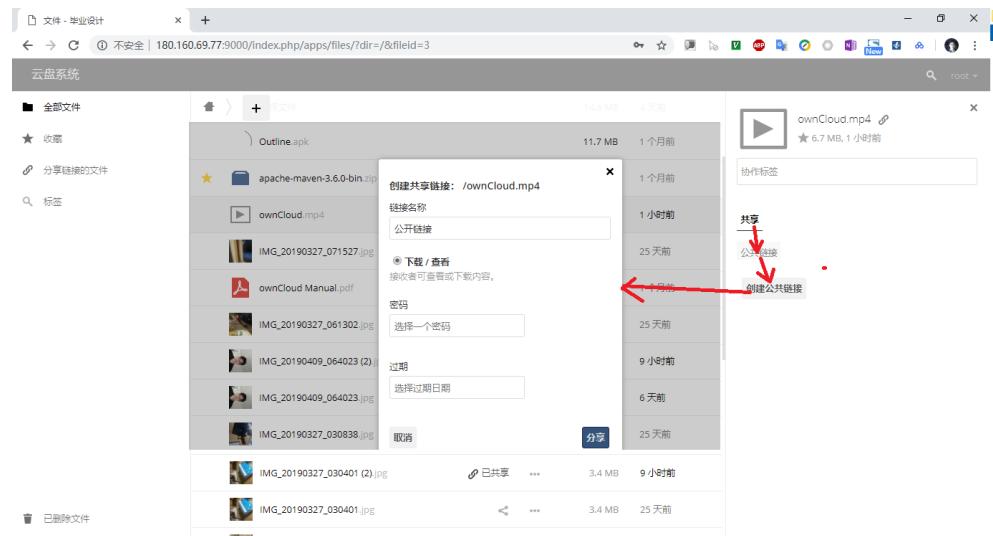


图 21: web 分享文件

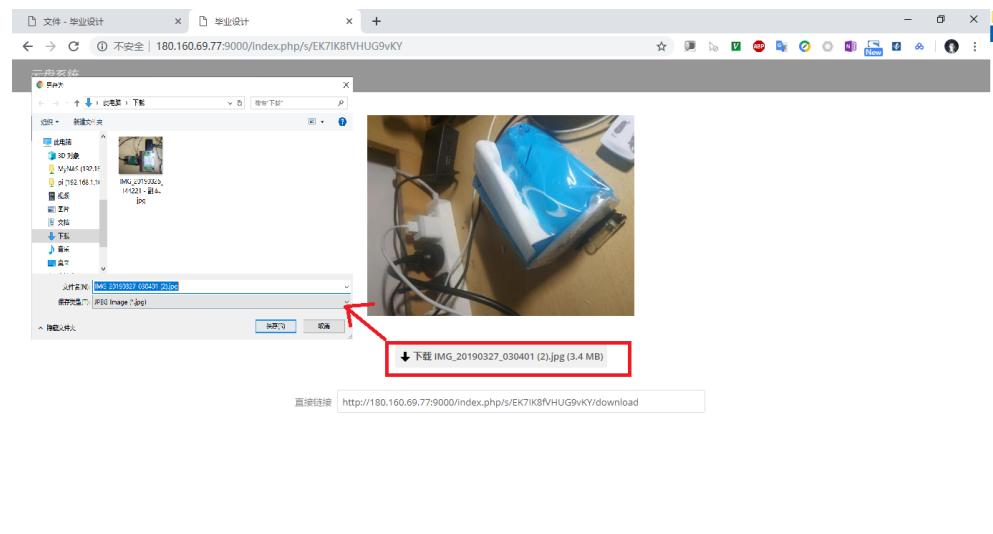


图 22: web 端访问分享文件



图 23: 安卓端分享文件

4.2.6 手机文件自动备份

安卓客户端默认不提供自动备份功能，因为自动备份的实现是安卓客户端会在后台开启一个 backup 的 service 线程，然后以固定的时间轮询查看备份目录是否用新文件增加，如果有增加，service 会调用 volley 网络请求工具上传文件给服务器，并在本地保存上传快照。

用户也可以选择是仅上传图片、仅上传视频、仅 WiFi 网络条件下上传、上传文件后是否保留原文件等设置项。



图 24: 安卓端文件备份设置

4.2.7 文件收藏、标签、搜索、已删除文件分类实现

系统会将文件分为全部文件、收藏、分享文件、已删除文件四大类，用户可以根据自己的需求为文件分类或者打标签，在 web 客户端会提供不通文件类型的入口，极大方便了用户查找文件。

实现的原理是创建四张文件表，用户在为文件归类时会往数据表中写入数据，所有文件默认时全部文件类型。当用户进入不同的文件入口时，服务器会各自查询 MySQL 中的数据表，返回结果会客户端。

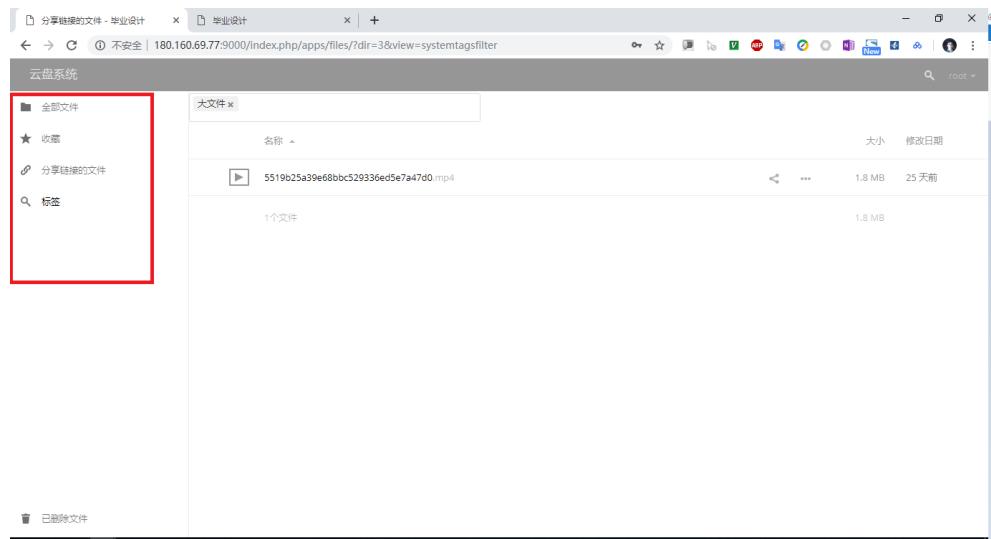


图 25: 安卓端文件备份设置

S

4.2.8 文件增量备份、异地备份实现

Merkle 树是一种基于哈希的数据结构。Merkle 树是树状数据结构，其中树中的每个叶节点是数据块，并且每个非叶节点是其子节点组合的散列。通常，Merkle 树是二叉树，这意味着 Merkle 树中的每个节点都有两个子节点。当然，Merkle 树可以是多叉树，例如以太坊平台。为简单起见，我们将在本文中仅讨论二进制 Merkle 树。

Merkle 树广泛用于分布式系统中进行数据验证^[28, 29]。通常在分布式系统中，由于我们将数据存储在许多不同的机器上，因此数据验证对于确保数据的可靠性和一致性尤为重要。例如，如果我们更新机器上的一段数据，则必须将更新传递给分布式系统中的所有机器以确保数据的最终一致性，以便比较不同机器上的数据。

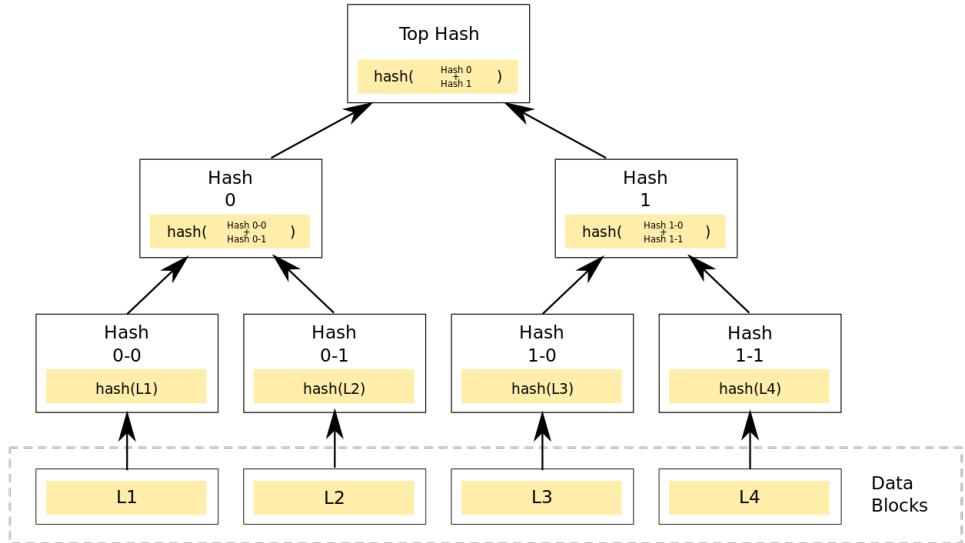


图 26: Merkle 树

s

本系统至少需要两台服务器进行数据备份，服务器间的数据必须保持同步。如果数据不一致，则必须快速定位不一致的节点。本系统为每台机器上的每个目录实现 Merkle 树，以便在比较两台机器之间的数据时，可以从 Merkle 树根节点开始比较。如果根节点相同，则表示两个副本当前一致，不再需要任何处理；如果不一致，则遍历 Merkle 树，可以快速定位到不一致的节点进行备份。相比于无脑地对文件进行完全备份，使用文件的哈希值作为文件的身份标识来构建 Merkle 树的做法提升了备份的速度和存储空间。

文件备份流程

- (1) A 服务器向 B 服务器发送需要与 Merkle 树的根节点对应的哈希值。
- (2) B 服务器接收此值并将其与正在构建的 Merkle 树的根节点哈希进行比较。
- (3) 如果两个值相同，则表示两者存储的文件相同。
- (4) 否则，B 服务器需要像 A 服务器那样的哈希值来请求与根节点对应的两个子节点。
- (5) A 服务器将相应的值发送给 B 服务器。
- (6) B 服务器比较相应节点中的散列值，并重复步骤 4 和 5，直到 B 服务器找到导致不同散列值的一个或多个数据块。

5. 系统测试

无论是软件开发完成后还是开发的过程中，验证需求，寻找 bug 都是不可避免的工作。一个软件系统需要通过专业测试工作，全面的对需求进行正确性验证，找出隐藏的 bug，并加以修复，才可以交付到用户手中使用。一个不经过测试的项目部署到生产环境中，出现影响用户体验甚至导致用户数据丢失、遭受黑客攻击等严重后果的概率很难被忽视。本章将对本人的测试工作进行详细的介绍，包括测试环境的搭建、测试计划、测试用例设计、测试执行以及测试结果数据的收集与评估等内容^[30]。

5.1 测试方法

软件测试方法主要需要验证系统的可靠性和故障恢复力、负载能力，抗压能力和并发性等。软件测试方法主要对系统的基本性能和服务器负载能力，服务器抗压性程序并发等情况进行测试，这需要验证系统的可靠性和失效恢复等功能，在系统测试阶段，除了传统的功能测试之外，还需要额外的系统测试来测试无线网络环境的性能和并发多任务处理性能。测试人员必须遵循测试要求，制定相应的测试方法和策略，创建相应的测试环境，并开发特定的测试用例。

(1) **黑盒测试** 黑盒测试也称为功能测试，测试系统功能是否能正常使用，黑盒测试可以说是最基本的测试。在测试中，测试人员可以不需要理解功能的实现细节、内部代码结构和内部特征，最直观的解释是就是点点鼠标，等待结果是否符合预期。它仅根据要求测试文档规范检查程序功能是否正常使用，也是测试项目的最开始的阶段，一般开发人员会在开发的过程中做一些简单的黑盒测试来验证代码的一般最低标准的正确性。黑盒测试侧重于程序的外部结构，无论内部逻辑结构如何，主要测试软件接口和软件功能。黑盒测试是测试人员给出测试用例，并根据用例输入相对应的数据和验证输出是否符合预期。显然，如果外部特征设计有问题或规格不正确，则无法找到黑盒测试方法。黑盒测试可以帮助我们发现下面的几类错误：

1) **功能不正确或遗漏** 比如安卓端在登录注册的过程中需要在本地持久化 JSESSIONID 来作为身份令牌来保持登录状态，不同于 web 端浏览器会自动在 Cookie 中保存 JSESSIONID，本系统的安卓端网络工具使用 Retrofit2 + RxJava + Okhttp3，是没有自动化保存 JSESSIONID 的功能的，在后来的功能测试工作中发现了缺乏 JSESSIONID 造成的验证错误。

2) **界面错误** 比如 Loaderunner 在测试过程中会保存用户界面的 html ui 界面，可以通过 Loderunner 设置验证条件来验证界面页面元素的可用性，命名是否统一，界面中的文字是否正确，文字和图像组合是否完美，页面是否漂亮。

3) **输入和输出错误** 最常见的测试场景就是用户在操作一些表单数据的时候，比如用户在登录时，输入不存在用户名，过长的用户名，不正确的口令、只输入用户名或口令、恶意用户注入 SQL 代码，或者 JavaScript 代码等，在这些过程中程序是否能给出人性化的错误提示都是要考量的一些方面。

4) **数据库访问错误** 例如，攻击者通过常规网页将其 SQL 代码传递给应用程序。从而通过执行开发人员未预期的 SQL 代码，偷窃或者破坏数据库中的数据。当然开发这也可以通过限制访问数据库账号权限、参数化使用命令、调用存储过程等方法来避免 SQL 注入。

5) **性能错误** 当存在大量并发方案和磁盘读写时，性能瓶颈是不可避免的。例如，磁盘 I/O 性能，CPU 性能和内存性能问题，分为网络瓶颈，服务器操作系统瓶颈，服务器硬件瓶颈和应用程序瓶颈。

(2) **压力测试** 软件压力测试是测试工作中一个非常重要的组成部分，一般在开发的初级阶段会做一些简单的功能测试，验证程序是否能正确跑起来，但是如果做了正规的压力测试的化，那么在生产环境中最最容易出现问题，最常见的就是系统在处理并发场景时遇到的性能瓶颈，常见的就是内存不足，数据库发生死锁等场景，如果没有在测试结果发现以上问题，那么在项目部署到生产环境上去之后就会遭受损失。和功能测试比较依赖于手动测试相比，性能测试会比较依赖自动化的一些操作，测试人员会编写一些测试脚本对并发场景、混合场景进行模拟，比较常见的性能测试工具是惠普的 LoaderRunner，测试人员可以先手动点击一些功能项，LoaderRunner 会将这些操作记录下来转化脚本，测试人员只需要进行简单的配置就可以模拟并发场景、混合场景了。同时 LoaderRunner 会提供一些测试报告，大大提高了测试人员的工作效率。性能测试主要对内部内存、CPU 可用性、磁盘空间和网络带宽这些方面进行测试。

5.2 测试环境和测试工具

本文需要特定的计算机硬件，软件，网络设备和日志数据平台来完成软件测试任务。稳定和可控的测试环境有助于测试人员缩短完成测试用例所需的时间，同时测试人员也不用额外费心设计测试用例和维护测试程序，并且可以保证所有提交的缺陷可以终准确无误的复现。

为了尽最大努力模拟真实情况，测试的过程是选择了寝室的局域网、校园网、无线蜂窝 4G 网络、北京上海城域网之间进行，测试用的硬件环境和软件环境、测试工具分别在下列三个表中列出。

下表是测试所需要的软件环境，包括 Linux、Apache、Mysql、PHP、Chrome、Internet、Android、Windows。下表是测试所需要的硬件环境，包括 TP-Link 路由器、Raspberry Pi、西部数据 2T 硬盘、计算机、移动手机 下表是测试所需要的压力测试工具，Web 端

表 8: 软件环境（相关软件、操作系统等）

名称	版本	数量	获得途径
Linux	Raspbian GNU/Linux 9	1	https://www.raspberrypi.org/
Apache	Apache/2.4.25 (Raspbian)	1	http://www.apache.org/
Mysql	mysql 5.7	1	https://www.mysql.com/
PHP	PHP 7.0.33	1	https://www.php.net/
Chrome	73.0.3683.86 (正式版本) (64 位)	1	https://www.google.com/chrome/
Internet Explower	IE 11.379.17763.0	1	Windows 系统自带
Android	MIUI 10/Android 9.0	2	手机厂商自带
Windows	windows 7/10	5	计算机厂商自带

Pi、西部数据 2T 硬盘、计算机、移动手机 下表是测试所需要的压力测试工具，Web 端

表 9: 硬件环境（网络、设备等）

名称	版本	数量	获得途径
TP-Link 路由器	TL-WR886N	1	自行购买
Raspberry Pi	Raspberry Pi 3 Model B+	1	自行购买 https://www.raspberrypi.org/products/
西部数据 2T 硬盘		1	自行购买
计算机	Thinkpad x260 / DELL 台式机	2	自行购买/公共财产
移动手机	小米华为	2	自行购买

使用 loadrunner, 安卓端用到了 Appium, 使用抓包工具 Wireshark 进行网络数据包分析。

表 10: 测试工具

名称	版本	数量	获得途径
Appium	V1.8.1	1	通过网络下载
Loaderunner	3.141.59	1	通过网络下载
Wireshark	3.0.0	1	通过网络下载

5.3 系统功能测试

在明确系统的测试目标之后可以根据测试目标制定合理的测试计划。就本云盘系统来说，测试目的主要包括系统实现是否满足云存储需求、性能是否足够稳定、以及安全性是否可靠三个方面。

表 11: 用户登录功能测试

测试项	操作步骤	预期结果	结果
正确输入	在登录框输入正确的用户名和口令，回车或点击登录按钮	登录成功后，web 端跳转到 index.php/apps/files 页，安卓端会跳转到 APPActivity 页面	通过
错误输入	在登录框输入错误的用户名口令，回车或点击登录按钮	提示登录失败，提供修改口令链接	通过
用户注册	在登录框输入未注册用户，提示请先注册，然后进行登录	提示登录失败，提供注册链接，注册成功后成功跳转	通过
用户注销提示	在登录框输入已经注销的用户，回车或点击登录按钮	提示用户不存在或口令错误，提供修改口令或注册用户链接	暂未实现
口令显示	在登录框输入口令	安卓端和 web 端对明文口令进行隐藏显示	通过
特殊字符	在登录框用户名输入中文、特殊字符，回车或点击登录按钮	登录成功后，web 端跳转到 index.php/apps/files 页，安卓端会跳转到 APPActivity 页面	通过
字符串长度	在登录框输入长度为 2, 4, 8, 16, 32, 64, 128, 256 长度用户名，回车或点击登录按钮	用户名在 4-32 长度之间允许向服务器发送验证请求，否则提示用户名过长或过短，如果用户自行构造登录请求，服务端验证后返回 400Bad Request 验证码	通过
口令特殊字符	在登录框输入中文，特殊字符口令，回车或点击登录按钮	提示口令不支持该格式，用户不能发送请求，如果用户自行构造登录请求，服务端验证后返回 400Bad Request 验证码	通过
口令长度	在登录框输入长度为 2, 4, 8, 16, 32, 64, 128, 256 长度口令，回车或点击登录按钮	口令长度在 4-16 之间用户可以发送请求，其余范围提示口令过长或过短，用户不能发送请求，如果用户自行构造登录请求，服务端验证后返回 400Bad Request 验证码	通过
口令大小写	在登录框分别输入大小写格式口令，回车或点击登录按钮	系统对口令大小写敏感，服务器使用 md5 对明文口令进行 hash 生成 128 位散列值，口令大小写不同会生成不同的散列值，和数据库口令进行比较返回验证结果	通过
常见口令	在登录框输入一些简单常用字符串口令，回车或点击登录按钮	登录成功后，web 端跳转到 index.php/apps/files 页，安卓端会跳转到 APPActivity 页面	通过
口令加密存储	登录注册成功后查看数据库口令存储方式	系统将用户明文口令进行 md5 哈希后存入数据库	通过

5.3.1 用户登录功能测试

用户登录功能是否能实现是保证本系统是否能正常使用的前提，因此在开发本云盘系统的时候，用户登录模块不仅仅是用户输入用户名和口令，在后台做验证，返回结果

表 12: 用户登录安全性测试

测试项	操作步骤	预期结果	结果
验证拦截	不登录状态，在浏览器地址栏直接输入需要登陆后地址、通过Http 请求的工具 Postman 发送登陆后地址请求	服务器会对请求地址请求进行拦截，返回 302 状态码和重定向地址 http://180.160.68.247:9000/index.php/login?redirecturl=请求地址	通过
Cookie httponly	web 端通过 js 代码获取浏览器 Cookie 值，安卓端通过 wireshark 对 app 请求进行截获，察看 Cookie 请求头	js 无法获取 Cookie 值，wireshark 无法查看 Cookie 值	失败
用户名和口令加密	通过 wireshark 对用户登录请求进行截获，查看用户名和口令是否通过加密的方式发送给 Web 服务器	用户的用户名和口令都得到加密，wireshark 看到的是加密后的密文	未实现
前后端协同验证	在前端用 js 验证输入参数是否符合规范，后端也需要验证	前端都进行参数验证，既能保障普通用户输入的正确性，也能防止黑客进行恶意攻击	通过
SQL 注入	通过 SQL 注入脚本工具在登录框构造 SQL 语句，比如万能登录 SQL 语句、删除表数据 SQL 语句	前后端都有进行输入验证，屏蔽 SQL 注入攻击，服务端验证后返回 400Bad Request 验证码	通过
XSS 攻击	通过 XSS 攻击脚本工具在登录框构造 js 代码，读取、篡改、添加、删除用户敏感数据	进行 Token 验证、Referer 验证、隐藏令牌等措施，防御 XSS 攻击	未实现
登录次数限制	通过 python requests 库对用户名和口令发动穷举攻击	服务器对单一 ip 地址的登录请求次数进行限制，登录次数超过 5 次会对该用户进行登录限制，5 小时内不能再进行登录，对于频繁发送请求进行 dos 攻击的 ip 地址加入黑名单，返回 401 状态码	通过
多用户单机器登录	在一台计算机或者移动终端上登录多个用户	用户可以通过浏览器的无痕模式进行单机器多用户登录，并且服务器会根据不通的 sessionid 返回对应的用户资源。安卓端可以通过 app 分身功能同时登录多个用户，服务器会根据不通的 sessionid 返回对应的用户资源	通过
单用户多机器登录	在 web 和移动 app 上同时登录	服务器会根据 sessionid 在后端保存登录用户信息用户可以在在 web 和移动 app 上同时登录	通过
验证次数限制	用户连续输入三次错误口令	用户一段时间内不允许登录，超出时间后能够继续登录	通过
session 日期	关闭浏览器、或者卸载重装 app、超过 24 小时发送访问资源请求	session 无效后用户无法访问系统资源	通过
输入方式	在用户名和口令输入框撤销、复制、粘贴文本	支持用户快捷键操作，比如撤销、复制、粘贴等操作	通过
用户同时登录	web 和 app 同时登录，同时上传下载文件	是允许同名用户同时登录进行操作，但是对文件进行加锁机制，保证事务的一致性，原子性	通过
网络判断	在断网条件下使用 app 客户端、web 客户端，在恶劣网络条件下使用 app 客户端、web 客户端	有未联网和网速较慢提示	通过

表 13: 文件上传下载测试表

测试项	操作步骤	预期结果	结果
命名检查	分别构造长度、后缀名符合或不符合规范的文件进行上传。	符合规范上传成功，否则失败	通过
类型检查	上传所有类型文件、二进制文件、可执行文件、文本文件等等。	所有文件上传成功。	通过
空文件上传	选择一个空文件，进行上传。	文件上传失败。	通过
正常小文件上传	分别选择 128KB、1MB、5MB、50MB 大小文件上传	文件上传成功。	通过
正常大文件上传	分别选择 100MB、200MB、500MB、1GB、2GB、4GB、5GB 大小文件上传	文件上传成功。	通过
超大文件上传	选择 6GM 大小文件上传如进行上传。	文件上传失败。	通过
正常小文件下载	分别选择 128KB、1MB、5MB、50MB 大小文件下载	文件下成功。	通过
正常大文件下载	分别选择 100MB、200MB、500MB、1GB、2GB、4GB、5GB 大小文件下载	文件下成功。	通过
上传下载响应时间	分别上传或者下载文件，测试响应时间	响应时间在 2s 以下	通过

就行，而且背后的实现考量的因素很多，因为作为最基本的系统功能，同时还要考虑到安全性，在开发的过程中就很不容易了。因此如表11所示，本文在正确输入、错误输入、用户注册、用户注销提示、口令显示、特殊字符、字符串长度、口令特殊字符、口令长度、口令大小写、常见口令、口令加密存储 12 个功能做了登录的功能测试。

测试结果表明，以上 12 个功能均已实现，用户在登录会有输入检测和结果提示，同时系统也对一些用户的输入进行验证，一定程度上保证了系统的安全性。

5.3.2 用户登录安全性测试

登录的安全性测试不同于登录功能上的一些验证，要验证登录功能是否在有一定的安全措施，需要理解功能实现的背后逻辑。安全测试需要用到 Wireshark 抓取请求包，分析 http 报文是否会泄露表单数据，比如用户名和口令在表单中是否以明文形式存储；同时要查看系统后台是否对口令进行哈希处理也需要在数据库查询，还有在实现单机单用户多机登录服务器的不同的 sessionid 的 session 中是否持有相同的用户信息，这些需要在后台打断点调试才能知道结果。如表12所示，本文分别在验证拦截、Cookie httponly、用户名和口令加密、前后端协同验证、SQL 注入、XSS 攻击、登录次数限制、多用户单机器登录、单用户多机器登录、验证次数限制、session 日期、输入方式、用户同时登录、网络判断 14 个进行验证。

测试结果表明，系统在 XSS 攻击、传输报文加密两个方面还存在欠缺，但是在其余安全措施方面均能达到预期目标。

5.3.3 文件传输测试

本系统的核心是为用户提供文件上传和下载的功能，整个系统的性能表现取决于树莓派作为功能承载的硬件，树莓派的网卡速率能达到 500Mb/s，同时在寝室的路由器也是 500M 网卡，实际测试下来在局域网能保证 2.5MB/s 的上行下载速度，在外网的环境下能保证 500KB/s 的上行下载速度，也比较符合中国电信的 20Mb 宽带套餐，因此传输速度方面实在是没有可以更多方面的工作的介绍，不同于集群的工作能力，本系统所能达到的性能表现只取决于树莓派的单机性能表现。同时在功能测试方面，本文分别对命名检查、类型检查、空文件上传、正常小文件上传、正常大文件上传、超大文件上传、正常小文件下载、正常大文件下载、上传下载响应时间 9 个方面进行测试，测试结果表明系统功能能达到设计目标。

5.3.4 性能测试

同在 Linux 系统中可以使用 dstat 监视系统资源利用率，在性能测试为 2 个并发用户在下载大型文件的场景下，如图27所示，树莓的 cpu 利用率达到 36 就直观感受来说，基本可以满足 2 个并发用户的上传、下载、分享、搜索文件的需求。

usr	sys	idl	wai	hig	sig	read	writ	recv	send	in	out	int	csw
5	4	55	36	0	1	41M	0	3920B	1195k	0	0	6383	4361
10	5	50	34	0	0	41M	0	5106B	1267k	0	0	6768	4716
9	5	39	47	0	1	41M	0	9310B	910k	0	0	5995	4073
3	3	48	46	0	0	39M	0	9156B	1381k	0	0	6786	4654
6	5	32	56	0	0	41M	0	4720B	1496k	0	0	6955	4904
6	5	31	57	0	1	41M	0	8160B	1514k	0	0	6685	4839
6	6	35	52	0	1	41M	0	11k	1587k	0	0	6989	4911
6	7	43	43	0	1	40M	64k	9932B	1342k	0	0	6003	4724
4	4	33	58	0	1	40M	0	7304B	669k	0	0	5585	3791
5	7	34	53	0	0	40M	0	7236B	580k	0	0	5612	3813
6	7	44	42	0	1	41M	0	9884B	1349k	0	0	6991	5249
6	4	38	51	0	0	41M	0	8964B	860k	0	0	6075	4416
6	4	45	44	0	1	40M	24k	8048B	657k	0	0	5563	3722
4	8	45	43	0	1	41M	0	7848B	617k	0	0	5737	3924
5	7	48	39	0	2	40M	0	6900B	851k	0	0	5732	3884
6	5	43	45	0	1	41M	0	8787B	809k	0	0	5583	3781
4	6	44	45	0	0	40M	0	5424B	495k	0	0	5033	3271
5	4	37	53	0	1	41M	0	8108B	1071k	0	0	6291	4375
6	3	38	52	0	1	40M	8192B	5416B	576k	0	0	4751	3004
4	5	37	54	0	1	41M	0	6516B	1011k	0	0	5628	3680

图 27: 树莓派系统资源监视

6. 总结和展望

6.1 研究和工作总结

本系统是利用闲置的家庭宽带和淘汰的硬盘资源，在基于廉价的通用小型计算机树莓派（英语：Raspberry Pi）的基础上开发出一个为用户提供文件的备份、共享、存储、访问、搜索、管理等功能，满足多终端访问的在线云盘存储系统。

本文的前期工作是收集市场上的云盘服务的定价、访问速度、系统稳定性和树莓派单片机配置、定价的数据信息，通过比较价格，安全可用性等方面，来决定本云盘系统是否存在开发实现的现实意义。通过数据分析，在和市场上商业的云盘服务比较之后，本系统在特定条件下存在成本和性能上的优势。

本系统利用 LAMP（Linux-Apache-MySQL-PHP）来进行开发系统后端服务器开发，同时提供 web 客户端和安卓客户端来进行文件的访问和备份。

本系统的 web 端在主流的浏览器 Chrome、Edge、IE11、QQ 浏览器、360 安全浏览器、360 极速浏览器、百度浏览器、opera 浏览器、手机端的 uc 浏览器、小米系统浏览器等浏览器都可以正常访问。在安卓客户端用户可以上传手机上的图片、照片、文本，可以设置自动备份来自动备份手机中的文件。

基于树莓派和二手机械硬盘的云盘系统的开发在成本上具有巨大的成本优势，同时在特定网络条件下的文件访问备份效果不错，也是本文的特殊意义所在。

6.2 展望

本文实现的云盘系统虽然能够一定程度满足当前项目的需求，但是在易用性、安全性和定制化方面仍然存在改进和提高。

易用性还需进一步改进。一个非计算机专业的用户要使用本系统，需要掌握在路由器网络端口配置、树莓派系统搭建、LAMP 环境搭建和使用、计算机网络知识、Linux 磁盘挂载、一些基本 Linux 命令和 vim 文本编辑、Android 程序编译等等专业知识。因此如果需要将本文的云盘系统作为产品推广，需要将一些操纵进行封装，让用户脱离复杂的参数配置而直接一键安装使用。如果仅仅提供源码和文档，我相信对于大部分用户，甚至是计算机专业的用户都会非常头痛，因此一个完善可直接使用的软件环境是今后需要努力的方向。

安全性还需进一步改进。因为开发时间仓促，本系统的网络传输协议使用的是 HTTP，我们知道 HTTP 是明文传输，因此很容易遭遇中间人攻击，传输数据被窃听和数

据内容篡改，虽然我系统的使用场景在寝室、家中的局域网内，遭受黑客攻击的可能性微乎其微，但是本系统有提供外网访问功能，因此还是存在一定的安全风险，所以将系统从 HTTP 迁移到 HTTPS 也是今后努力的方向。

定制化还需进一步改进。本系统提供了两种客户端 Web 和 Android，但是有电脑用户更倾向于使用桌面的客户端进行数据备份下载，市场上主流的云盘服务百度云就同时支持 Web，手机端，桌面端的应用，人们也有使用桌面软件的使用习惯，因此下一步的改进的方向是开发出桌面客户端和 IOS 应用，从而覆盖更广的用户。当然本系统还有一些缺失的功能，比如动态分享、垃圾文件清理、通话记录备份、通信录备份、短信备份等功能还存在缺失，因此今后功能的添加也是一个努力的方向。

参考文献

- [1] 中国信息通信研究院. 中国公有云发展调查报告. 2018. <http://www.caict.ac.cn/kxyj/qwfb/ztbg/201808/P020180817600313812960.pdf>.
- [2] 中国互联网络信息中心. 第 42 次《中国互联网络发展状况统计报告》. 2018. <http://www.caict.ac.cn/kxyj/qwfb/ztbg/201808/P020180817600313812960.pdf>.
- [3] 艾媒咨询. 2016 年中国个人云盘行业研究报告. 2016. <https://www.iimedia.cn/147842005961632451.pdf>.
- [4] WU J, PING L, GE X, et al. Cloud Storage as the infrastructure of cloud computing[J]. IEEE, 2010: 380–383.
- [5] ARMBRUST M, FOX A, GRIFFITH R. A view of cloud computing[J]. Communications of the ACM, 2010, 53(4): 50–58.
- [6] 周可, 王樺, 李春花. 云存储技术及其应用[J]. 中兴通讯技术, 2010(4): 24–27.
- [7] 维基百科. Cloud storage. 2018. https://en.wikipedia.org/wiki/Cloud_storage.
- [8] 百度百科. NAS 网络存储. 2019. <https://baike.baidu.com/item/NAS%E7%BD%91%E7%BB%9C%E5%AD%98%E5%82%A8/8856025?fr=aladdin>.
- [9] STERLING T, ANDERSON M, BRODOWICZ M. High Performance Computing[M/OL]. Ed. by STERLING T, ANDERSON M, BRODOWICZ M. [S.l.]: [s.n.], 2018: 549–578. <http://www.sciencedirect.com/science/article/pii/B9780124201583000186>. DOI: <https://doi.org/10.1016/B978-0-12-420158-3.0.00018-6>. ISBN: 978-0-12-420158-3.
- [10] ORLANDO A, AMATO P, CARACCIO D, et al. Linux page fault analysis in android systems[J/OL]. Microprocessors and Microsystems, 2019, 66: 10–18. <http://www.sciencedirect.com/science/article/pii/S0141933118302254>. DOI: <https://doi.org/10.1016/j.micpro.2019.01.006>. ISSN: 0141-9331.
- [11] ABBOTT D. Chapter 3 - Introducing linux[J/OL]. 2018: 29–54. <http://www.sciencedirect.com/science/article/pii/B9780128112779000031>. DOI: <https://doi.org/10.1016/B978-0-12-811277-9.00003-1>.
- [12] RaspberryPi.org. Raspberry Pi — Teach, Learn, and Make with Raspberry Pi. 2019. <https://www.raspberrypi.org/>.
- [13] 李新宇. 网络云盘介绍—以 360 云盘和百度云为例[J]. 无线互联科技, 2014, 1: 23.
- [14] 贺博, 王欣. 个人云存储市场发展状况分析[J]. 现代电信科技, 2012, 42(10): 11–14.
- [15] 京东. 【西部数据 Elements 新元素系列】西部数据 (WD)2TB USB3.0 移动硬盘 Elements 新元素系列 2.5 英寸 (稳定耐用海量存储)WDBUZG0020BBK【行情报价价格评测】-京东. 2019. <https://item.jd.com/6772447.html>.

- [16] Apple. iCloud storage plans and pricing - Apple Support. 2019. <https://support.apple.com/en-us/HT201238>.
- [17] Dropbox. Choose the right Dropbox for you and your business. 2019. <https://www.dropbox.com/plans>.
- [18] Amazon. Amazon S3. 2019. <https://amazonaws-china.com/cn/s3/>.
- [19] Google. Google Drive. 2019. <https://mashable.com/category/google-drive/>.
- [20] 百度. 首页-会员中心-百度网盘. 2019. <https://pan.baidu.com/buy/center#/svip>.
- [21] Microsoft. Microsoft OneDrive Plans. 2019. <https://onedrive.live.com/about/en-US/plans/>.
- [22] XIANGFENG C, LINSHAN B, TONGYUN C. Investigation on the Electrochemical-Mechanical Polishing of NiP Substrate of Hard Disk[J/OL]. Rare Metal Materials and Engineering, 2011, 40(11): 1906–1909. <http://www.sciencedirect.com/science/article/pii/S1875537212600125>. DOI: [https://doi.org/10.1016/S1875-5372\(12\)60012-5](https://doi.org/10.1016/S1875-5372(12)60012-5). ISSN: 1875-5372.
- [23] GISCHEL B. 9 - Data backup[J/OL]. 2016: 485–510. <http://www.sciencedirect.com/science/article/pii/B9781569904985500101>. DOI: <https://doi.org/10.3139/9781569904992.009>.
- [24] 维基百科. LAMP - 维基百科，自由的百科全书. 2019. <https://zh.wikipedia.org/wiki/LAMP>.
- [25] 维基百科. Java Platform, Enterprise Edition. 2019. https://en.wikipedia.org/wiki/Java_Platform,_Enterprise_Edition.
- [26] 维基百科. .NET Framework. 2019. https://en.wikipedia.org/wiki/.NET_Framework.
- [27] GitHub. Clxy/BigFileUploadJava: Upload big file by java. 2019. <https://github.com/clxy/BigFileUploadJava>.
- [28] BERMAN P, KARPINSKI M, NEKRICH Y. Optimal trade-off for Merkle tree traversal[J/OL]. Theoretical Computer Science, 2007, 372(1): 26–36. <http://www.sciencedirect.com/science/article/pii/S0304397506008693>. DOI: <https://doi.org/10.1016/j.tcs.2006.11.029>. ISSN: 0304-3975.
- [29] SHOUFAN A, HUBER N, MOLTER H G. A novel cryptoprocessor architecture for chained Merkle signature scheme[J/OL]. Microprocessors and Microsystems, 2011, 35(1): 34–47. <http://www.sciencedirect.com/science/article/pii/S014193311000044X>. DOI: <https://doi.org/10.1016/j.micpro.2010.07.002>. ISSN: 0141-9331.
- [30] 维基百科. System testing - Wikipedia. 2019. https://en.wikipedia.org/wiki/System_testing.

致谢

至此，我的本科毕业论文工作落下帷幕，我在华东师范大学计算机科学与软件工程学院的四年学习生涯也即将画上句号。这四年中，给予我关怀与帮助的人太多，谨在此处向他们表达最衷心的感谢。囿于篇幅和文笔，无法表达出我全部的感激之情。

首先我最想感谢的就是我的指导老师。您的本科专业课为我打开了密码学的大门，您扎实的学术功底，认真的治学态度令人敬佩。在我的论文工作中，从开题到交稿，您给予了我巨大的帮助，始终不厌其烦的为我答疑解惑。即使平时的科研和教学工作都很繁忙，您仍然在抽空，甚至是占用自己的休息时间对我进行指导。对此我深表愧疚

其次我要感谢我的同学，尤其是可爱的室友们。你们在学习和生活中给予我温暖和帮助，每个人身上都有无数值得学习的地方，你们是我大学四年最宝贵的收获。

最后是要感谢我的家人，你们是我最坚强的后盾，也是我努力学习的不竭动力。