**基于hadoop的云盘系统设计文档**

**一、环境搭建**

整合SpringMVC、Spring、Spring data、前端easyUI（暂定）的web开发环境，构建云盘maven项目

**二、数据库设计**

云盘系统涉及到两种数据库，一个是关系型数据库mysql，另一个是非关系型数据库Hbase，目前设计Hbase中只存放文件信息，其他信息存放在mysql中。数据库设计如下：

用户登录信息表yun\_auth

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 说明 |
| id | varchar(32) | 主键 |
| username | varchar(64) | 登录名 |
| password | varchar(32) | 密码，加密后的（统一采用MD5加密） |
| createTime | datetime | 首次入库时间 |
| updateTime | datetime | 最近更新时间 |
| createIp | varchar(32) | 创建时的IP |
| latest\_loginTime | datetime | 最后一次登录时间 |
| latest\_loginIp | varchar(32) | 最后一次登录时的IP |
| state | int | 状态，0：未激活；1：正常；2：禁用 |
| extend | Varchar(32) | 扩展字段 |
| person\_id | varchar(32) | 外键，引用yun\_person表的主键 |

个人信息表yun\_person

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 说明 |
| id | varchar(32) | 主键 |
| realname | varchar(64) | 真实姓名 |
| email | varchar(64) | 电子邮箱 |
| phone | varchar(64) | 手机号码 |
| introduction | varchar(4000) | 个人简介 |
| picture | varchar(128) | 个人头像，存放路径 |
| createTime | datetime | 首次入库时间 |
| updateTime | datetime | 最近更新时间 |
| extend | varchar(32) | 扩展字段 |

上传文件表yun\_upload\_file

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 说明 |
| id | varhcar(32) | 主键 |
| name | varchar(256) | 文件原始名称 |
| size | long | 文件大小 |
| type | varchar(128) | 文件类型 |
| path | vachar(128) | 文件路径，存放HBase中某条记录的rowkey |
| person\_id | varchar(32) | 外键，引用yun\_person表的主键 |
| createTime | datetime | 首次入库时间 |
| updateTime | datetime | 最近更新时间 |
| order | int | 排序字段 |
| uploadIp | varchar(32) | 上传文件时的IP |
| dir\_id | varchar(32) | 外键，引用yun\_dir表的主键，表示该文件所在的目录 |
| state | int | 文件状态 |
| extend | varchar(32) | 扩展字段 |

文件目录表yun\_dir

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 说明 |
| id | varchar(32) | 主键 |
| name | varchar(128) | 目录名 |
| parent\_id | varchar(32) | 父目录的ID |
| person\_id | varchar(32) | 外键，引用yun\_person表的主键 |
| createTime | datetime | 首次入库时间 |
| updateTime | datetime | 最近更新时间 |
| extend | varchar(32) | 扩展字段 |

文件分享表yun\_share\_file

|  |  |  |
| --- | --- | --- |
| 字段名 | 类型 | 说明 |
| id | varchar(32) | 主键 |
| upload\_file\_id | varchar(32) | 外键，引用yun\_upload\_file的主键 |
| person\_id | varchar(32) | 外键，引用yun\_person的主键 |
| code | varchar(32) | 分享的唯一编码 |
| state | int | 分享状态 |
| type | int | 分享类型，0：私密分享；1：公开分享 |
| createTime | datetime | 首次入库时间 |
| updateTime | datetime | 最近更新时间 |
| extend | varchar(32) | 扩展字段 |

Hbase中存放表

文件信息表yun\_file

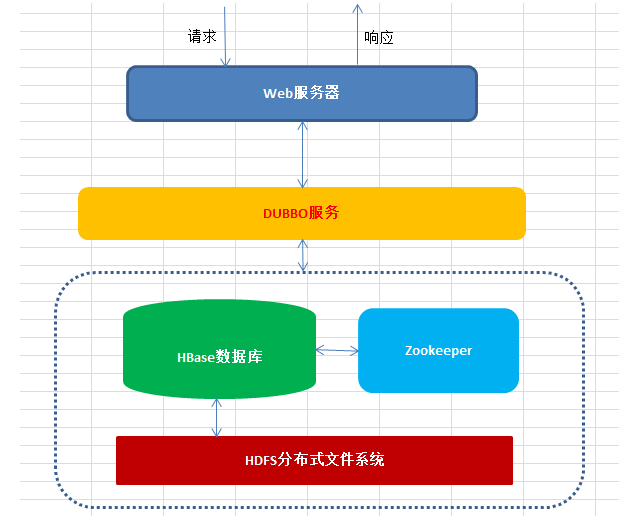
|  |  |
| --- | --- |
| 字段名 | 说明 |
| row\_key | row\_key值 |
| file\_info\_file | File\_info为列族，存放文件 |
| file\_info\_person.id | File\_info为列族，存放个人信息表(yun\_person)的主键 |

**三、搭建Hadoop、Hbase、zookeeper环境**

由于我们需要hadoop的hdfs分布式文件系统，所以需搭建hadoop生态环境

**四、系统架构及技术选型**

4.1系统架构：



其中以HDFS分布式文件系统作为底层文件系统，在此之上搭建Hbase分布式数据库，同时Hbase依赖zookeeper进行协同作业，以此构建了Hadoop生态圈。Hadoop具有很容易进行水平扩展，健壮性强，可靠性高的特点。

Web系统不是直接与Hbase连接，而是将操作Hbase的方法注册到Dubbo中，提供对外服务，web系统是调用Dubbo中的服务，遮掩可以更好的进行服务治理，同时解耦。

4.2技术选型：

1. HDFS，作为底层分布式文件存储系统
2. Hbase，作为分布式数据库，将文件存到Hbase中
3. Zookeeper作为Hbase协同作业，同时作为Dubbo注册中心
4. SpringMVC+Spring+Spring data jpa，作为web系统的开发组合
5. Mysql，关系型数据库
6. easyUI，前端框架

**五、需求确立**

5.1个人管理中心  
a、注册  
b、登录，或者通过社交平台关联登录  
c、修改密码

5.2上传文件  
a、能够支持超大文件上传  
b、上传文件能够指定上传到某个目录

c、能够创建文件夹

5.3文件归档  
a、增删改查文件夹和文件  
b、支持移动文件

5.4文件回收站  
a、删除的文件归档的位置  
b、回收站中的文件可以恢复  
c、删除回收站中的文件，将永久性删除

5.5下载文件

5.6预览文件  
a、 对于PDF、txt文件、视频文件等可以提供预览功能

5.7分享文件  
a、公开分享，直接点击链接就可以看到分享的文件  
b、私密分享，点击链接后，需要输入密码才能看到文件

5.8 每个注册用户可以免费获得一定的容量，同时支持扩容

5.9整个系统必须支持高可用、高并发、高吞吐量，可扩展

**六、系统开发及部署**