

基于Hadoop的云盘存储系统设计与实现

徐翔, 张光亚

(广东海洋大学教育信息中心, 广东 湛江 524088)

摘要:云存储目前已被广泛地应用于多个领域。通过数据的远程存储与备份,用户可远程随处办公。Hadoop作为分布式文件系统具有高扩展性、高吞吐量和存储海量数据等特性。文章提出了一种基于Hadoop存储平台的云盘系统的设计思路,研究利用HDFS作为云盘存储平台的可行性,搭建了以SpringBoot框架为基础的后台服务器并对接Hadoop集群系统,解决云盘系统在存储性能上、存储安全性上所遇到的问题,为同类系统的开发提供了重要参考。

关键词:Hadoop;HDFS;云盘;分布式存储

中图分类号:TP317 **文献标识码:**A

文章编号:1009-3044(2023)03-0078-04

DOI:10.14004/j.cnki.ckt.2023.0135

1 引言

随着目前互联网产业的飞速发展,以及通讯设备应用的普及,计算机在人们日常生活中发挥着越来越重要的作用,信息的处理、传输和采集已然成为现代信息技术的三大基石^[1]。在未来,随着大数据、物联网、人工智能、5G等技术的不断突破与发展,全球的数据量将越来越大,信息的存储和管理变得复杂,单机硬盘存储已满足不了用户的文件存储和管理需要。

为了适应目前信息化社会对于支持网络和多种数据信息的应用软件的要求,以网络存储为核心的个人云盘存储系统应运而生。云盘存储相对于传统的实体磁盘来说更方便快捷,用户不需要把储存重要资料的实体磁盘带在身上,而可以通过互联网轻松地从云端读取自己所存储的信息,解决了单机硬盘存储量少、管理效率低、网络传输能力差等有关于文件存储的问题。

在云存储服务面向用户使用之前,用户数据的存储存在一定的局限性,例如当用户需要向个人存储系统中存储海量数据文件时,因系统本身没有实现大数据文件优化存储处理的方案,原本的系统存储性能将会很大程度地降低。此时即使对存储服务器进行扩展升级,存储性能也不会有明显的提升。因此,相对于传统的存储服务,云存储无论是对于需要更大存储空间的个人,还是对于寻求高效异地数据备份解决方案的企业都越来越受欢迎。云存储提供了一种以安全可靠的方式存储和迁移数据的方式,它允许个人和企业将他们的文件存储在云服务提供商处,以便他们能够在网络设备上对个人文件进行访问。

基于Hadoop^[2-4]的HDFS是谷歌文件系统的开源实现项目,项目成立之后便逐渐成为各个企业或机构

的云存储系统解决方案,同时也是当前在云存储领域最重要的研究对象。HDFS^[5]在Hadoop集群中担任着集群数据存储重要角色,具有高稳定性与可扩展性等优点,并且能够运行在廉价机器上,一定程度上降低了云存储系统设计的成本。国内互联网巨头百度和阿里巴巴分别在2006年和2012年就对Hadoop进行关注并研发使用,解决了大数据云存储的相关问题,为用户提供了底层的模型计算以及存储服务。

基于Hadoop的HDFS分布式文件系统是当前云存储领域的研究热点^[6-8]。本课题研究并设计以Hadoop为基础的大数据云盘存储系统具有重要意义。

2 云盘系统总体架构

本系统的总体架构分为三部分:客户端、业务逻辑层及大数据存储集群,具体如图1所示。

用户从客户端中向后台服务器发起用户注册与

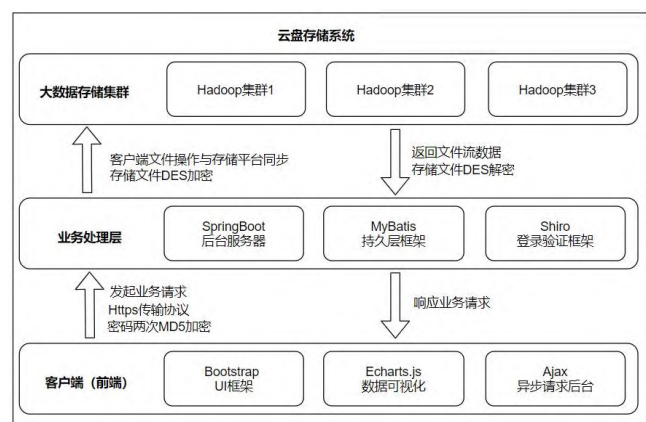


图1 云盘系统总体架构

收稿日期:2022-07-28

基金项目:广东海洋大学“创新强校”工程重大科研与培育项目(No:q18305)

作者简介:徐翔(1974—),男,硕士,高级实验师,广东海洋大学教育信息中心网络管理部主任,研究方向为计算机网络通信及信息安全;张光亚(1970—),男,硕士,副研究员,广东海洋大学教育信息中心主任,研究方向为教育信息化建设。

登录请求,验证通过后可以在个人云盘空间内管理自己的目录与文件,可以对目录或文件进行新增、删除、重命名等操作。用户向后台服务器发起的所有请求都需经过安全性的处理,以防网络攻击者窃取用户的个人隐私信息。

业务处理层则用于处理用户发起的具体请求,若业务处理的过程中存在逻辑错误,则会向用户弹出错误提示框,让用户规范自己的相关操作。对于用户上传的文件数据,后台服务器负责将其转化为流数据,并经过加密处理后存入大数据存储集群中。用户的文件存储数据的相关记录将存储至MySQL数据库内。

在大数据存储集群中,由于HDFS集群具有副本机制,每个Hadoop节点都会分别存储一个文件的副本,即一个文件有3个副本,这些副本可以根据特定的算法分配到三个Hadoop节点中,一定程度上避免了数据丢失的情况。故大数据存储集群必须使用三个或三个以上的服务器,从而实现Hadoop全分布式的集群环境。

3 云盘系统实现

3.1 前端 Web 页面设计

云盘系统的前端界面采用基于Bootstrap的UI框架,实现用户登录注册界面及云盘系统模板界面。Bootstrap是一个开源的、基于HTML5及CSS3和JS的响应式布局框架,因此它可以很好地兼容电脑端页面与手机Web端页面的布局。在本系统中,考虑同时设计电脑端页面及手机端页面,方便用户在多种形式的设备上使用本系统。

用户在前端对云盘进行的一系列操作将使用经过定制后的Ajax技术,用于异步请求后台资源,并返回固定格式的用户提示框。为了方便用户能够在线读取特定格式的文件内容(如文本、图片等),还引入了基于Bootstrap的文件读取框架。对于数据分析模块,为满足数据可视化需求,引入了较为热门的Echarts.js组件,用以生成数据图表信息。

3.2 后台 Web 服务器设计

云盘系统采用SpringBoot作为后台Web服务器基础框架,并使用MVC模式将后台服务器分为接口层(Controller)、业务逻辑层(Service)和数据链路层(Mapper)三层架构,完成整个系统的数据传输和交互。MyBatis是一个持久层框架,用于建立数据库的访问链接,在此过程中开发者只需关注SQL语句的书写逻辑,方便快捷,且能与SpringBoot完美集成。系统中还采用了Shiro用户登录安全验证框架,它能够轻松地与JavaSE和JavaEE进行集成,用于用户登录时的认证、授权、加密、会话管理等功能。

3.3 数据库设计

数据库概念设计如图2所示,包含用户、目录和文件实体。每个用户可以创建多个目录及上传多个文件,但是一个目录或文件只能属于一个用户,因此用户与目录、文件之间都是属于一对多关系。

为了统计用户的相关操作信息以及操作云盘时所产生的流量数据,还需设计日志表实体与流量表实体,实体关系如图3所示。日志实体主要的字段为所

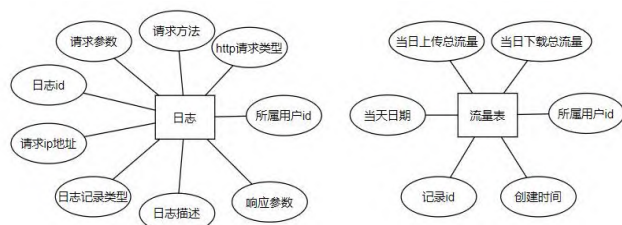


图2 日志、流量表E-R图

属用户ID和响应参数,用于统计用户的相关操作参数,同时也包括用户上传与下载的流量使用参数,日志表为流量统计提供了相关的数据源。流量表实体则是用于记录数据分析后的流量信息,主要字段有所属用户ID、当日上传总流量以及当日下午总流量等。

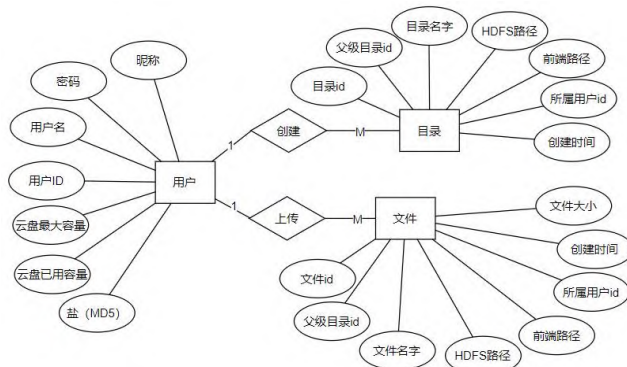


图3 用户、目录和文件实体关系图

3.4 基于MapFile的小文件存储

HDFS设计之初就是为了存储大容量文件,并没有对小文件的存储进行相关的优化。在Hadoop2.0版本之后,HDFS中的每个文件存储数据块以128MB为单位,而用户个人数据文件(以文档、图片等为主)的容量大小远远小于128MB的数据块阈值,这些小文件额外占用了HDFS中更多的容量空间,造成存储空间的浪费。当用户数量增大时,HDFS便迎来了大量小文件存储所造成的存储效率问题。

为了解决HDFS不适合存储小文件的问题,提出了使用基于MapFile的方案来优化小文件的存储,提高存储效率。MapFile基本上由两大部分组成,分别是用于存储数据的Data块,以及存储索引文件的Index块。在使用MapFile方案存储小文件时,文件数据将会被分为一个键值对,该键值对中的“键”指的是序列化后的文件名,“值”则是指文件本身的内容。多个文件数据的键值对数据会被合并为一个文件存储在Data块中,与此同时建立每个文件数据的映射关系,在Index索引块中记录每个文件数据的键值对信息。这样一来,在通过MapFile访问小文件时,可以通过Index索引块快速定位到相关的文件内存位置,提高小文件检索效率。

4 系统测试

4.1 测试环境搭建

4.1.1 Hadoop 集群搭建

为了对系统进行测试,搭建了具有三台服务的Hadoop集群。服务器采用Linux系统,基本配置相同。为

了快速搭建 Hadoop 集群,首先创建一个模板服务器,然后通过拷贝的方式复制另外两个相同配置的服务器。

模板服务器名称为 Hadoop102,默认安装有 Java JDK8 与 Hadoop3.1.3 两个必要组件。另外两台服务器名称分别为 Hadoop103 和 Hadoop104。将 Hadoop102 中的 JDK 与 Hadoop 组件复制给这两个新服务器。为了实现跨服务器传输文件,采用 shell 语言的 rsync 命令,其具体作用是把服务器指定目录的所有文件拷贝到另一个服务器。最后,把 Hadoop 的重要运行节点分配到三台服务器中。在 Hadoop 全分布式环境中,为了保证集群正常运行,需要启动五个基础节点:NameNode(管理节点)、DataNode(工作节点)、Secondary-NameNode(NameNode 副本节点)、ResourceManager(资源管理节点)以及 Nodemanager(监控资源节点)。其中,每个服务器都必须拥有一个 DataNode 与一个 Nodemanager 节点,以保证每个服务器能够正常运行相关程序。因此,对于另外三个重要节点 NameNode、SecondaryNameNode 以及 ResourceManager,需要分别分配到三个服务器中。服务器的节点分配情况如表 1 所示。

表 1 Hadoop 集群节点分配

服务器 组件	Hadoop102	Hadoop103	Hadoop104
HDFS	NameNode DataNode	DataNode	SecondaryNameNode DataNode
YARN	NodeManager	ResourceManager NodeManager	NodeManager

为了分配服务器节点,需要在各服务器的 core-site.xml 文件中配置 NameNode 节点的主机地址,并指定 NameNode 启动在 Hadoop102 服务器的 8020 端口上。NameNode 节点关键属性配置如下所示:

```
01. <!-- 【1】指定NameNode的地址,配到Hadoop102 -->
02. <property>
03.   <name>fs.defaultFS</name>
04.   <value>hdfs://hadoop102:8020</value>
05. </property>
```

在每台服务器的 hdfs-site.xml 文件中配置 SecondaryNameNode 节点的主机地址,并指定 SecondaryNameNode 节点启动在 Hadoop104 服务器的 9868 端口上。关键属性配置如下所示:

```
01. <!-- 【2】2nn web端访问地址,也相当于指定了2nn运行服务器-->
02. <property>
03.   <name>dfs.namenode.secondary.http-address</name>
04.   <value>hadoop104:9868</value>
05. </property>
```

在每台服务器的 yarn-site.xml 文件中配置 ResourceManager 节点的主机地址,并指定 ResourceManager 节点启动在 Hadoop103 服务器上。关键属性如下所示:

```
01. <!-- 【3】指定ResourceManager的地址-->
02. <property>
03.   <name>yarn.resourcemanager.hostname</name>
04.   <value>hadoop103</value>
05. </property>
```

经过以上合理配置的 Hadoop 集群即使有一台服务器发生了故障,其他两台服务器仍然保留着重要的数据信息,最大程度上避免数据丢失。

4.1.2 后台 Web 服务器搭建

后台服务器将以 SpringBoot 作为基础框架,以 Ma-

ven 作为项目管理工具。当一个基于 Maven 的项目创建完成后,需要使用 Maven 仓库将相关依赖属性添加到 pom.xml 文件内。为了让 Maven 项目引入 Spring-Boot 框架,则需要向 pom.xml 文件中添加如下所示的基础依赖。

```
01. <!-- SpringBoot基础依赖 -->
02. <dependency>
03.   <groupId>org.springframework.boot</groupId>
04.   <artifactId>spring-boot-starter-jdbc</artifactId>
05. </dependency>
06. <dependency>
07.   <groupId>org.springframework.boot</groupId>
08.   <artifactId>spring-boot-starter-web</artifactId>
09. </dependency>
10. <dependency>
11.   <groupId>mysql</groupId>
12.   <artifactId>mysql-connector-java</artifactId>
13.   <scope>runtime</scope>
14. </dependency>
15. <dependency>
16.   <groupId>org.springframework.boot</groupId>
17.   <artifactId>spring-boot-starter-test</artifactId>
18.   <scope>test</scope>
19. </dependency>
```

引入成功后,使用 Java 开发工具启动 SpringBoot 项目,后台服务器就已具备了基础的 Web 服务器请求与响应的功能。为了使后台服务器具备上文“技术架构设计”中所描述的相关技术,还需要添加 Mybatis 框架、shiro 框架、Hadoop 客户端等依赖,具体的依赖属性配置如下所示。

```
01. <!-- mybatis-plus -->
02. <dependency>
03.   <groupId>com.baomidou</groupId>
04.   <artifactId>mybatis-plus-boot-starter</artifactId>
05.   <version>3.4.3.1</version>
06. </dependency>
07. <!-- shiro框架 -->
08. <dependency>
09.   <groupId>org.apache.shiro</groupId>
10.   <artifactId>shiro-spring</artifactId>
11.   <version>1.7.0</version>
12. </dependency>
13. <!-- Hadoop客户端 -->
14. <dependency>
15.   <groupId>org.apache.hadoop</groupId>
16.   <artifactId>hadoop-client</artifactId>
17.   <version>3.1.3</version>
18. </dependency>
```

后台服务器搭建还需要配置 HTTPS 请求协议,为此首先申请 SSL 证书。目前较为成熟的云计算服务器腾讯云或阿里云可进行免费申请。获取证书后,将证书相关文件导入 SpringBoot 项目中,随后通过向 application.yml 文件添加相关配置来完成协议的导入。具体导入代码如下所示:

```
01. ssl:
02.   key-store: classpath:mysslfilecn.pfx
03.   key-store-password: 000000
04.   key-store-type: PKCS12
```

最后,为了解决 HTTP 协议兼容问题,设置 HTTP 页面重定向到 HTTPS 协议页面中。

4.2 系统实现效果

设计了测试用例对系统登录、新建目录、上传文件、删除文件、下载文件、文件夹重命名等功能进行了测试,系统实现效果如图 4 所示。

4.3 系统流量统计

为了测试系统的稳定性,对系统进行了为期 8 天的上线测试,测试用户数量为 2-10 之间的随机值,系统流量统计情况如图 5 所示。用户对云盘文件的操作记录将会记录到数据库中的日志表内,当需要进行数据分析时,后台服务器会对日志表的数据进行筛选



图 4 系统上传文件实现效果图

整理并导出包含序号、用户 ID、日期、上传流量和下载流量五列字段的数据集,随后再将数据集上传至 HDFS 集群中执行 MapReduce 数据合并分析。分析完毕后,输出结果将自动写入 MySQL 数据库中的“流量统计表”内。

5 结束语

本文采用分布式文件系统 Hadoop 设计并实现了云盘存储系统,可实现云端文件的管理和存储功能,包括新建文件夹、删除文件夹、上传文件、下载文件、文件重命名等。本文详细介绍了系统架构、前端页面设计和后台服务器搭建等设计与实现细节信息,为同类系统开发提供了参考。

参考文献:

[1] 林子雨. 大数据技术原理与应用:概念、存储、处理、分析与



图 5 系统流量统计分析

应用[M].2版.北京:人民邮电出版社,2017.

- [2] Shvachko K, Kuang H R, Radia S, et al. The hadoop distributed file system[C]//2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST). May 3-7, 2010, Incline Village, NV, USA. IEEE, 2010: 1-10.
- [3] 翟永东. Hadoop 分布式文件系统(HDFS)可靠性的研究与优化[D]. 武汉:华中科技大学, 2011.
- [4] 刘慧敏. 基于 Hadoop 的小文件存取技术的研究[D]. 北京:北京工业大学, 2020.
- [5] 洪旭升, 林世平. 基于 MapFile 的 HDFS 小文件存储效率问题[J]. 计算机系统应用, 2012, 21(11): 179-182.
- [6] 程旺. 基于 Hadoop 的云存储系统的设计与实现[D]. 哈尔滨:黑龙江大学, 2020.
- [7] 王跃. 基于 Hadoop 分布式文件系统的分析与研究[J]. 计算机光盘软件与应用, 2011, 14(9): 161-162.
- [8] 王宏志, 李春静. Hadoop 集群程序设计与开发[M]. 北京:人民邮电出版社, 2018.

【通联编辑:王力】

(上接第 73 页)

与特定行业安全威胁分析,支持外部威胁的精准识别与追溯。和通报预警机制结合,实现网络安全事件的处置与响应。

(4) 基于微观与宏观的安全视角,使网络安全全局可视能力得到提高,对全网安全态势实时监控,通过宏观视角对整体安全情况进行掌控,通过微观角度对安全线索捕获,从而快速判断网络整体态势和威胁相关影响范围、目的和攻击路径,支撑有效响应和决策^[8]。

互联网安全防护是大数据环境下解决安全问题的重点,要维护互联网的安全。目前,维护互联网安全的重点为控制访问设置,利用用户访问权限的设置,对互连网络安全使用进行保证。比如,通过身份认证和密码避免黑客攻击,根据访问权限进行设置。其次,设置数据加密,隐藏用户数据信息,保证用户数据的安全性。为了加强网络安全,还能够设置网络隔离,实现用户数据存储系统的防火墙,对大量信息进行识别,筛选有效信息。设置防火墙能够对计算机隔离,对隔离后网络安全进行保证。最后,实现用户网络的入侵检测,及时阻拦非法用户入侵行为。此种监控技术使用主动方式对网络安全防御,消除防火墙不足,也是全方位网络化安全实时保护的技术^[9]。

4 结束语

在信息化时代下,网络安全成为国家和人们所重视的问题。在现代异构网络模式中,网络安全监控难

度比较大,只有创建健全、科学、完全的异构网络安全监控体系,才能够保证监测信息的可靠性与准确性,提高监测效率,及时发现网络安全问题。相信在研究人员的努力下,寻找更加先进的关联算法,从而准确、快速地将数据转变成有用的信息知识,为人们提供可靠的网络信息数据。

参考文献:

- [1] 张伟华,王海英. 基于 B-CNN 模型的异构网络大数据知识扩充算法研究[J]. 兵器装备工程学报, 2022, 43(6): 290-294.
- [2] 刘蓓, 禄凯, 程浩, 等. 基于异构数据融合的政务网络安全监测平台设计与实现[J]. 信息安全研究, 2020, 6(6): 491-498.
- [3] 闵小翠. 基于 K-均值聚类的多模态异构大数据检测算法研究[J]. 电子设计工程, 2020, 28(14): 63-67.
- [4] 杨木伟, 肖辉, 党超辉, 等. 基于智慧广电“安全大脑”的广播电视网络安全防护体系建设研究与应用[J]. 广播电视网络, 2022, 29(6): 54-58.
- [5] 李峻屹. 基于大数据的 K-means 聚类算法的网络安全检测应用研究[J]. 机械设计与制造工程, 2021, 50(9): 115-118.
- [6] 董彦俊, 李泽峰, 陈大海. 基于马氏距离的异构网络异常大数据剔除方法[J]. 计算机仿真, 2022, 39(1): 408-411, 445.
- [7] 张涛涛. 基于异构数据源的网络安全监测平台设计和实现[J]. 网络安全技术与应用, 2022(6): 6-8.
- [8] 陈海倩. 基于移动互联网大数据的异构实时计算架构分析[J]. 自动化技术与应用, 2020, 39(9): 44-47.
- [9] 陈霞霞, 王小龙, 蒋太交, 等. 基于多源异构大数据挖掘的流感病毒防控预测预警平台构建研究[J]. 中国生物工程杂志, 2020, 40(S1): 109-115.

【通联编辑:代影】