毕业设计（论文）开题报告

基于 OpenHarmony 的分布式文件去重系统设计与实现

一、课题的目的及意义

（一）课题的目的及意义

1．背景

随着人类生活的智能化和社会信息技术的发展，全球数据量面临着无限制地扩展和增加。传统的存储已满足不了市场的需求，虽然有技术成熟、性能良好、可用性高等优点，但面对海量数据，缺点也越来越明显：如扩展性差、成本高、可靠性低等。而分布式存储的出现为高效安全的存储带来新的可能。

分布式存储是一种去中心化存储系统。分布式思想的出现可以追寻到 1965 年高登·摩尔（Gordon Moore）所提出的摩尔定律。经过几年的发展，人们发现摩尔定律的预测是符合现实的。这就意味着，集中式系统的运算能力每隔一段时间才能提升一倍。如果系统需承载的计算量的增长速度大于摩尔定律的预测，那么在未来的某一个时间点，集中式系统将无法承载所需的计算量。无论是要以低价格获得普通的性能，还是要以较高的价格获得极高的性能，分布式系统都能够满足。并且受规模效应的影响，系统越大，性价比带来的收益越高。进入到互联网快速发展的时期，分布式系统相比集中式系统的另一个更明显的优势：更高的可用性。例如，有 10 个能够承载 10000 流量的相同的节点，如果其中的 2 个出现问题，只要实际流量不超过 8000，系统依然能够正常运转。

但随着分布式存储系统逐渐投入到生产实践，许多问题也暴露出来。主要的问题之一是分布式节点间存在大量重复数据。这些重复数据的产生一方面会导致系统有效存储容量降低，另一方面重复数据还会增加系统维护的难度。总的来说，过多的重复数据会给系统带来很多危害，除非系统永远不需要维护，而通常这种情况不会发生，因此在系统开发过程中要尽力采取各种措施来避免数据重复。为了解决分布式系统中存在的文件重复问题，本课题将基于 OpenHarmony 平台完成分布式文件去重系统设计与实现的任务。

OpenHarmony 是由开放原子开源基金会（OpenAtom Foundation）孵化及运营的开源项目，目标是面向全场景、全连接、全智能时代，基于开源的方式，搭建一个智能终端设备操作系统的框架和平台，促进万物互联产业的繁荣发展。目前

Harmony OS 已经初步实现了这一目标，可以说是 openHarmony 的一个成功典范。课题将通过前期对 OpenHarmony 系统分析、开发技术分析、设计周期分析；系统架构设计、代码实现；智能开发套件上板测试、后期测试等必要开发步骤完成。

2．目的及意义

本课题具有很强实践意义和理论意义。课题迎合了当下由集中式存储转变为分布式存储的演变进程，以解决分布式存储中存在的实际问题，促进分布式存储的发展。其研究目的在于：

（1）减少分布式系统中的重复文件，提高磁盘的有效存储容量

（2）降低分布式系统数据维护的难度

其理论意义在于：

（1）提出了一种分布式文件粒度去重的架构

（2）提供了一种文件远程读写的实现方案

（二）研究现状分析

本课题受到了许多经典的分布式文件系统的启发，将从分布式文件的架构和演化去分析，梳理其中的关键技术和未来的发展方向。回顾发展简史，大致可以把分布式存储分为四个发展阶段：

表 1 分布式存储发展阶段

阶段	研究重点	主要代表
1980s 网络文件系统	实现网络环境下的文件共享，解决客户端与文件服务器的交互问题	AFS 文件系统、SUN 公司的 NFS 文件系统
1990s 共享 SAN 文件系统	解决存储系统的可扩展性和面向 SAN 的共享文件系统	IBM 研制的 GPFS
2000s 面向对象并行文件系统	研究重点主要集中在对象存储技术，如何进行高效的元数据管理和提高数据访问的并发性	Ceph 文件系统、Google 的 GFS 文件系统
2010s 云文件系统	EB 级大规模存储系统，数据高可用性方法，高效智能存储技术，以及新型的计算存储融合系统和应用感知存储。	目前很多分布式文件系统都在往的云的方向发展，如：GlusterFS、Ceph 等

随着云计算的发展以及新硬件，新算法的加持，一些创新的分布式文件系统也出现了，它们给这个方向提供了新的演进思路。

2003 年 Google 开发的 Google File System[1]（简称 GFS）问世。GFS 是 Google 公司为了满足本公司需求而开发的基于 Linux 的专有分布式文件系统，可以部署在廉价的商务服务器上，在保证系统可靠性和可用性的同时，大大降低了系统的成本。GFS 是典型的中心化架构设计，后来的很多分布式系统都参考这种架构，该模式的最大优点是便于管理，因为中心服务器可以获知所有子服务器的状态，因而可以很方便的得知各个子服务器的负载状况等。但是这一模式也有一个比较致命的缺点，那就是单点故障。当单点故障发生在中心服务器时，将导致整个系统的不可用。

HDFS[2](Hadoop Distributed File System)是基于 Google 发布的 GFS 论文设计开发的。HDFS 是 Hadoop[3]技术框架中的分布式文件系统，对部署在多台独立物理机器上的文件进行管理。HDFS 会将数据自动保存多个副本，通过增加副本的形式，提高容错性。其中一个副本丢失以后，可以自动恢复。但是在面对多副本而出现的大量重复数据时，HDFS 并没有对此进行优化。

Ceph[4]最早起源于 Sage 就读博士期间的工作、成果于 2004 年发表，并随后贡献给开源社区。经过十几年的发展，已成为应用最广泛的开源分布式存储平台。Ceph 相比其它存储的优势点在于它不单单是存储，同时还充分利用了存储节点上的计算能力，在存储每一个数据时，都会通过计算得出该数据存储的位置，尽量将数据分布均衡。同时由于 Ceph 的良好设计，采用了 CRUSH 算法[5]、HASH 环[6]等方法，使得它不存在传统的单点故障的问题，且随着规模的扩大性能并不会受到影响。

FastDFS[7]是一个开源的轻量级分布式文件系统。它解决了大数据量存储和负载均衡等问题。特别适合以中小文件为载体的在线服务，如相册网站、视频网站等等。但是 FastDFS 不支持 POSIX[8]通用接口访问，通用性比较的低。

本课题所实现的分布式文件去重系统具有架构简单、通用性高和去重效果明显的特点。系统设计借鉴了最经典 GFS 的中心化架构设计，将元数据存在在中心节点处统一管理，其他数据节点负责数据的重删和传输。这样设计的系统在实现上相对简单，易于扩展和维护，但也具有中心化架构易发生单点故障的缺点。同时本系统也基于标准文件接口规范实现远程文件读写的功能，通用性较高。在分布式系统安全方面，本课题参考其他分布式系统安全设计，使用了文件传输加密算法确保文件的安全传输。

分布式文件去重系统主要适用在小批量小文件传输型分布式系统，在大型分布式系统下可能并不适用。作为一个系统，永远是追求多个维度，多个指标的平衡，

最经典的 Google GFS 存在诸多的问题，但是不妨碍它成为 Google 存储的核心基础设施，HDFS 的诸多问题也不妨碍它在 Hadoop 生态中地位。同时我们也应该面向未来来看待问题，随着新的思想，新的硬件，新的场景的出现，通用的解决方案，放之四海而皆准的产品是不存在。

在以 5G、人工智能、物联网为首的新技术驱动的新时代，万物智连的数据形态本身就是高度分布式的存在。而随着技术应用落地的进一步发展，其产生的化学反应会加剧，新数据形态、新应用模式和新价值需求会进化得越来越快，企业也将更依赖数据采集、数据分析来提升生产力、开展业务。因此，更能适应大数据分析、视觉计算、多云[9]等场景的分布式存储，在未来取代传统存储方式是大势所趋。

表 2 经典分布式文件系统对比

名称	优点	缺点
GFS	架构简单；集群水平扩展 可以构建在廉价的服务器上 存储容量大；支持 PB 级存储规模；	适合大文件读写，不适合小文件存储；没有实现 POSIX 的标准文件接口；中心化架构易发生单点故障；
HDFS	多个副本容错性高；可以处理大数据（GB\TB 级）；可以构建在廉价的机器上；	不适合低延时数据访问； 无法高效对大量小文件进行存储；不支持并发写入；
Ceph	高扩展性:本身并没有主控节点，扩展起来比较容易;特性丰富:Ceph 支持对象存储、块存储和文件存储服务，故称为统一存储；	安装运维复杂； 加入了日志导致数据实际上会双写，性能不足； 整体代码质量一般；
FastDFS	主备 Tracker 服务，增强系统的可用性；支持在线扩容机制，增强了系统的可扩展性；支持主从文件，支持自定义扩展名	不支持断点续传，对大文件不友好；不支持 POSIX 通用接口访问，通用性较低；同步机制不支持文件正确性校验，降低了系统的可用性

二、课题任务、重点研究内容及实现途径

（一）课题任务

1.设计（论文）主要内容与要求如下：

（1）熟悉 OpenHarmony 操作系统；
（2）熟悉典型的分布式存储架构；
（3）掌握分布式文件去重系统的设计与实现方法；
（4）掌握完整项目的设计和实现方法；
（5）掌握网络编程的方法和程序调试方法；
（6）能够撰写专题论文。

2.课题的重点难点：

本题目涉及学科面较广，涉及到编译原理、操作系统、程序设计、数据结构等多门课程的知识。课题要求学生了解常见分布式系统的设计和 Linux 的文件读写等操作，了解各类常用数据结构，需要学生们有刻苦学习的精神，了解相关的操作系统和编译原理知识，以及相关的编程工具，能够快速查找相关资料以及挖掘、总结出有效信息的能力，以及团队协作的精神。通过本题目，将提升学生知识综合应用、文献查阅分析、系统分析设计与实现等能力。

3.开发环境：

Windows 和 Ubuntu 系统混合开发模式、使用 openHarmony 智能开发套件。

4. 课题必要工作：

（1）完成课题要求的基于 OpenHarmony 的分布式文件去重系统设计与实现；
（2）按照规范完成系统的开发和测试，体现良好的专业素养；
（3）按照要求完成毕业论文。

（二）课题重点研究内容

1.系统架构

本课题的重点之一在于系统架构的设计。课题实现的分布式文件去重系统借鉴了经典分布式系统的中心化架构设计，基于 Linux 平台进行实验，自主编写文件读写接口，通过文件指纹进行对比以确定需要删除的重复文件。本系统主要涉及到的模块如下：Metadata node、Data node、自定义动态链接库。

2.系统优化

系统的优化可以从两个方面入手：增强系统安全和提高传输并发量。

在分布式文件去重系统中涉及到文件传输时的安全性问题，如何保证用户文件数据在传输中的安全性是本课题的研究重点之一。本课题拟采用 Openssl 的文件加密传输技术确保用户在远程文件传输时的文件安全。

此外，在分布式文件去重系统中会出现多个设备同时对一个设备进行文件访问

的情况，这种情况要求分布式文件系统应当具备高的信息交互并发量。本课题预期使用 socket 高并发模型来达到多用户高并发访问的目的。

（三）课题实现途径

1.方案设计

中心化架构示意图如下：

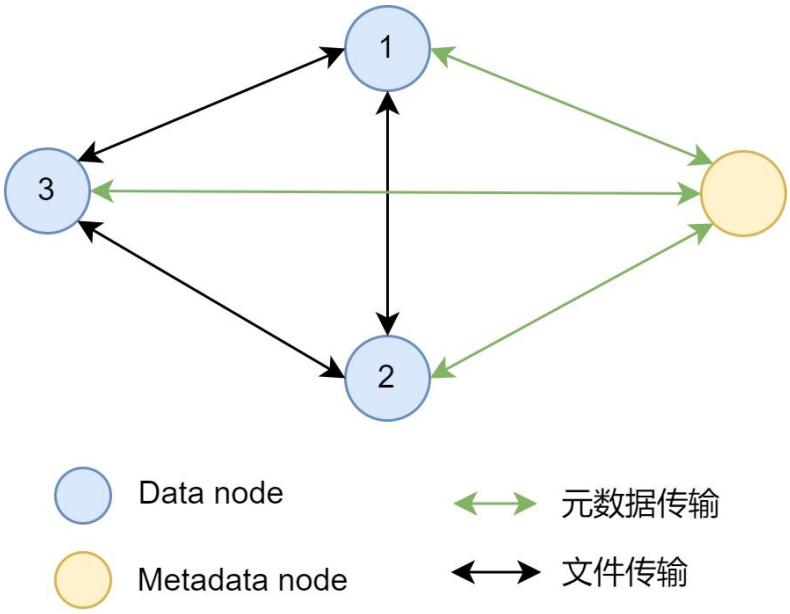


图 1　系统中心化架构图

（1）Metadata node 设计

Metadata node 又称作元数据节点。在分布式文件去重系统中主要负责文件元数据管理和重删指令的发送。在元数据节点中只会存储文件的元数据，不会存储文件数据本身。元数据节点需要存储全局文件唯一的指纹和地址，当系统中出现新增文件时，元数据节点会在全局指纹表中进行比较，以此来推断文件是否重复。此外元数据节点还负责 Data node 地址存储，方便出现重复文件后发送重删指令到特定地址的数据节点。

（2）Data node 设计

Data node 又称作数据节点。在分布式文件去重系统中数据节点主要负责保存文件数据和传输文件。数据节点会启动扫描线程定期扫描设备中的新增文件，计算新增文件的我指纹，并将新增文件的 inode 信息连同指纹打包成新增文件表发送至元数据节点。此外，数据节点既作为文件传输的服务器端，又作为文件接收的客户端，因此文件数据的传输与访问不会涉及到元数据节点。这样设计可以使得文件传输不经过中心节点中转，避免中心节点成为文件系统的瓶颈。

（3）自定义动态链接库

自定义动态链接库主要在 Data node 端使用。Data node 端出现读写文件的请求时需要调用自定义库中的文件读写接口，该自定义实现的接口符合 C 库标准读写接口规范，在 C 库标准读写功能基础上添加了远程读写的功能，可以实现多个 Data node 的远程文件读写需求。个别具体接口读写流程图如下：

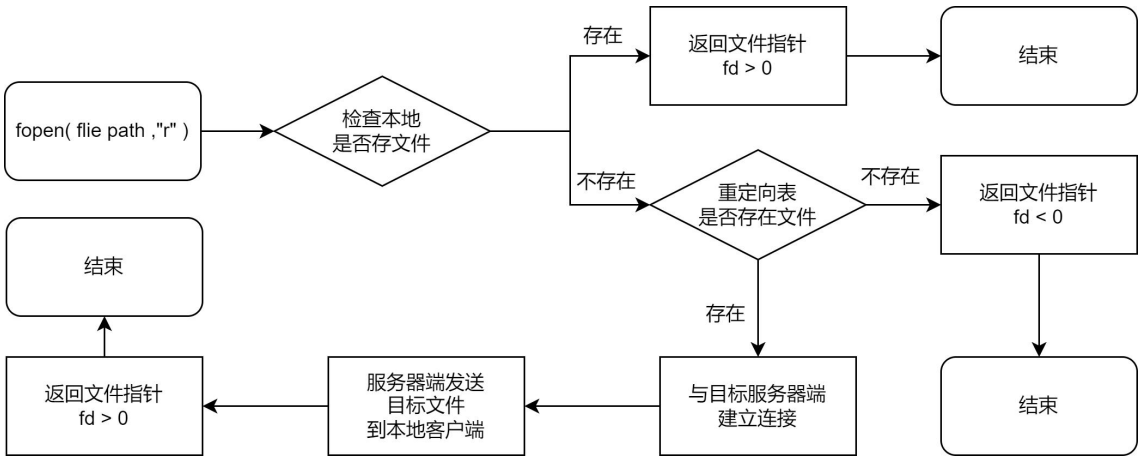


图 2 fopen 打开只读文件流程图

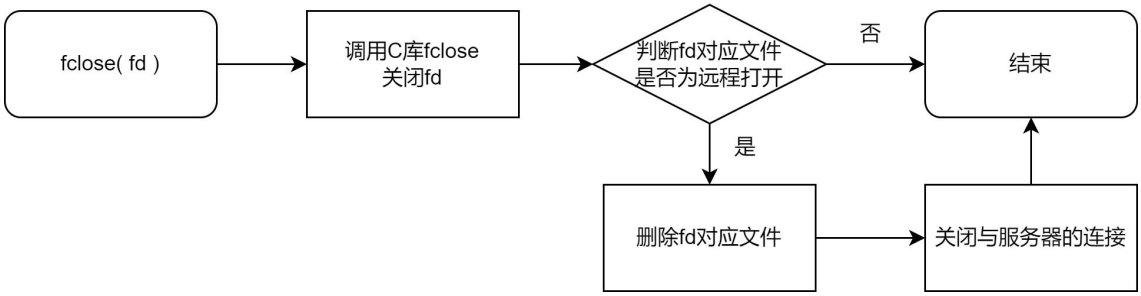


图 3 fclose 关闭只读文件流程图

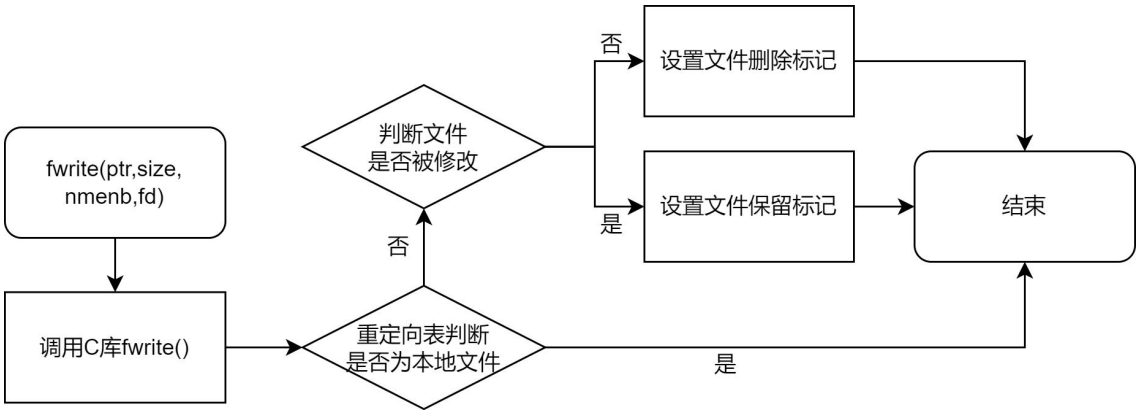


图 4 fwrite 写追加文件流程图

3.关键技术

（1）进程间通信技术[10]

Data node(数据节点端)需要在扫描进程和远程文件请求进程间使用进程间通信以共享文件重删表。进程间常见的通信方式有以下 5 种：管道 pipe、命名管道 FIFO、消息队列 MessageQueue、内存共享 SharedMemory 和信号量 Semaphore。本课题主要采用了内存共享技术实现进程间通信。共享内存就是映射一段能被其他进程所访问的内存，这段共享内存由一个进程创建，但多个进程都可以访问。共享内存是最快的 IPC 方式，它是针对其他进程间通信方式运行效率低而专门设计的。它往往与其他通信机制，如信号量，配合使用，来实现进程间的同步和通信。

（2）库打桩技术[11]

库打桩技术，可以截获对共享库函数的调用。该技术为 Linux 环境 C 语言独有。应用上可以如可以控制函数调用的输入输出值，以自己的逻辑替换函数调用。其基本思想是：创建一个与目标函数相同原型的包装函数，通过编译时函数实现的搜索机制、或链接时函数符号解析搜索的机制、或运行时动态链接库的加载机制，将自定义的包装函数替换目标函数。本课题基于库打桩技术独立编写了一套可以远程文件访问的动态链接库，用户使用该动态链接库编译代码可以实现设备的远程文件访问功能。

（3）文件加密技术

本课题在文件传输时使用了 openssl 库的文件加密传输技术。openssl 是一个安全套接字层密码库，囊括主要的密码算法、常用密钥、证书封装管理功能及实现 ssl 协议。Openssl 整个软件包大概可以分成三个主要的功能部分：ssl 协议库 libssl、应用程序命令工具以及密码算法库 libcrypto。此外 Data node（数据节点）会计算文件指纹，文件指纹使用的是 MD5 文件加密技术。 MD5 即 MessageDigest Algorithm 5（信息-摘要算法 5），用于确保信息传输完整一致。是计算机广泛使用的杂凑算法之一（又译哈希算法），主流编程语言普遍已有 MD5 实现。

MD5 算法具有以下特点：

- 压缩性：任意长度的数据，算出的 MD5 值长度都是固定的。
- 容易计算：从原数据计算出 MD5 值很容易。
- 抗修改性：对原数据进行任何改动，哪怕只修改 1 个字节，所得到的 MD5 值都有很大区别。
- 强抗碰撞：已知原数据和其 MD5 值，想找到一个具有相同 MD5 值的数据是非常困难的。

参考文献：

[1] GHEMAWAT S, GOBIOFF H, LEUNG S T. The Google file system. ACM SIGOPS Operating Systems Review[J]. 2003,37(5):29-43.

[2] KARUN A K, CHITHARANJAN K.A review on hadoop—HDFS infrastructure extensions[C]//2013 IEEE conference on information & communication technologies. IEEE, 2013: 132-137.

[3] GHAZI M R, GANGODKAR D. Hadoop, MapReduce and HDFS: a developers perspective[J]. Procedia Computer Science, 2015, 48: 45-50.

[4] Weil S A, Brandt S A, Miller E L, et al. CRUSH: Controlled, scalable, decentralized placement of replicated data[C]//Proceedings of the 2006 ACM/IEEE conference on Supercomputing. 2006: 122-es.

[5] AGHAYEV A, WEIL S, KUCHNIK M, et al. File systems unfit as distributed storage backends: lessons from 10 years of Ceph evolution[C]//Proceedings of the 27th ACM Symposium on Operating Systems Principles. 2019: 353-369.

[6] 一致性 Hash 算法[EB/OL][2017-12-01].http://blog.csdn.net/wudiyong22/article/details/78687246.

[7] LIU X, YU Q, LIAO J. FastDFS: a high performance distributed file system[J]. ICIC express letters. Part B, Applications: an international journal of research and surveys, 2014, 5(6): 1741-1746.

[8] BUTENHOF D R. Programming with POSIX threads[M]. Addison-Wesley Professional, 1997.

[9] 分布式云[EB/OL].http://www.ibm.com/cn-zh/topics/distributed-cloud.

[10] BRYANT R, O'HALLARON D. Computer Systems A Programmer's Perspective(深入理解计算机系统 修订版) [M].龚奕利，雷迎春,译.中国电力出版社.

[11] BOVET D, CESATI M.Understanding the Linux Kernel[M].O'Reilly Media , 2005-11.

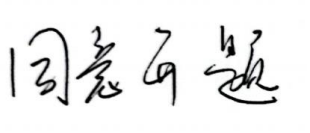
三、进度计划

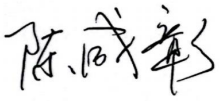
序号	起止周次	工　作　内　容
1	1 周至 4　周	学习和熟悉开发环境； 上网和在数字图书馆查阅相关文章；
2	1 周至 2　周	系统总体设计； 配置环境搭建系统框架/平台；
3	3 周至 5　周	接口设计/模块设计；文件去重算法设计；
4	6 周至 8　周	接口/模块代码实现； 程序调试；
5	9 周至 11　周	程序烧板；系统测试； 设计改进/功能完善；
6	12 周至 14　周	编写技术文档，完成毕业论文初稿；
7	14 周至 16　周	完善毕业论文；进行答辩；

学生签名： 

2023　年 2　月 28　日

四、指导教师意见



指导教师签名： 

2023　年 2　月　28　日

说明：

1．开题报告应根据教师下发的毕业设计（论文）任务书，在教师的指导下由学生独立撰写。

2．本页不够，请加页。