面向云存储的存储网关①

马 军, 石 辉, 裴文斌, 王福利

(西安雷迪维护系统设备有限公司, 西安 710065)

摘 要: 对大容量数据存储和快速读写的需求与计算机网络技术的发展, 使得网络化存储系统成为网络服务器系统中 I/O 子系统研究的热点, 作为网络存储系统的关键部件, 对分布式文件系统的研究具有非常重要的意义. 目前开源社区提供了 KFS、moosefs、FastDFS、TFS、GFS[1]等多款分布式文件系统, 其中多数提供了基于 Linux 操作系统的 API 或者存储网关, 却没有提供 Windows 版的存储网关. 主要对分布式文件系统 Windows 版存储网关的设计框架和思路进行介绍与分析, 并实现了一个基于 HDFS 的 Windows 版分布式文件系统的存储网关程序 dfsclient.

关键词: 分布式文件系统; 存储网关; HDFS; dfsclient

Storage Gateway for the Cloud Storage

MA Jun, SHI Hui, PEI Wen-Bin, WANG Fu-Li

(Xi'an Leidi Maintenance System Equitment Co. Ltd, Xi'an 710065, China)

Abstract: With the development of the computer and network technology, the growing demands for the huge data storage and the high-performance data access make the storage systembasedon network becoming a research hotspot, especially the research of I/O subsystem about network server system. The research about distributed file system as a key part in the network server system is meaningful. Currently open source community provides several DFS(distributed file system) projects, such as KFS, moosefs, FastDFS, TFS, GFS. Most of projects support only API or storage gateway on Linux system, but not giving support API or storage gateway on Windows system. This study introduces the designing frameworks and the thinking process aboutstorage gateway on Windows system, and then gives a specific example program based on HDFS—dfsclient.

Key words: distributed file system; storage gateway; HDFS; dfsclient

随着计算机网络的发展与网络带宽的不断增长, 利用网络技术来提高存储系统的容量、可靠性与可扩展性成为可能. 近年来, 分布式网络存储已经成为存储技术发展的新趋势. 分布式文件系统可以将分散在网络中的存储资源组织起来, 构成大容量的虚拟磁盘存储空间. 该系统能够更好地解决多客户端并发操作下, 有限的网络带宽和磁盘 I/O 对数据访问速度的限制; 同时, 由于该系统在重复利用原有设备、动态扩充虚拟空间容量、容灾备份等多方面具有的优越性, 因而受到了开源社区和相关技术公司的追捧, 并在很短的时间内出现了很多开源的分布式文件系统, 如 KFS、moosefs、FastDFS、Ceph、HDFS、GFS 等, 以及闭源的分布式文件系统 LoogStore、BWFS[2]等.

就开源分布式文件系统而言, 多数不支持在 Windows 操作系统上进行应用开发. 即使某些分布式文件系统支持, 也只是提供些动态库或者静态库, 这对已经存在于 Windows 操作系统上的原有应用程序而言, 不但要考虑重新编码, 还要考虑当前应用程序和分布式文件系统提供的动态库或静态库之间的接口和语言兼容性等问题. 假如分布式文件系统的接入网关能够以网络虚拟磁盘的方式提供服务, 这样既可以减少程序之间的耦合性, 又可以避免对原有应用程序的

①收稿时间:2014-03-26;收到修改稿时间:2014-07-18

修改. 本文主要对分布式文件系统的 Windows 网络虚拟磁盘的实现方式进行研究, 并实现了一个 Windows 版存储网关程序 dfsclient.

1 基于Linux存储网关的实现分析

基于 Linux 系统的存储网关实现方法有两种: 一种实现方式如 Open-ISCSI, 其所有功能都在内核态实现, 用户态只完成网络连接管理和磁盘设备的挂接操作. 实质是在网络设备对象 net_device 上抽象出一个虚拟的磁盘驱动设备对象, 然后通过一个网络连接对象 sock 将网络设备对象和虚拟磁盘设备关联起来, 以实现网络虚拟磁盘功能. 当将这个磁盘设备挂接到根文件系统上的时候, 就将这个设备对应的文件系统挂接到根文件系统上了, 对挂载点内文件的操作通过网络对应为对远端存储设备的操作.

另一种实现方式是通过用户空间文件系统 (Filesystem in Userspace, 简称 FUSE), 如 ZFS、glusterfs、luster 和 moosefs 的存储网关都是通过 fuse 的方法实现的.

传统上操作系统在内核层面上对文件系统提供支持, 但通常内核态的代码难以调试, 生产率较低. 在用户空间实现文件系统能够大幅提高生产率, 简化了为操作系统提供新的文件系统的工作量, 特别适用于各种虚拟文件系统和网络文件系统. 但是, 在用户态实现文件系统必然会引入额外的内核态/用户态切换带来的开销, 对性能会产生一定影响. 不过, 相对于减少软件之间的耦合性, 避免原有应用程序的二次开发而言, fuse 具有很大优越性.

其运行机制如下图 1 (虚框部分表示一个独立的任务).

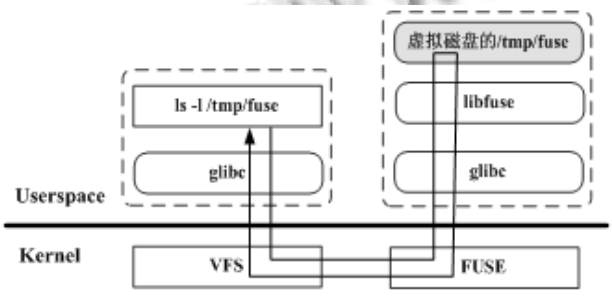

图 1 fuse 运行机制示意图

2 基于Windows存储网关的设计与实现

基于 Windows 系统存储网关的实现方法和基于 Linux 系统存储网关的实现方法相似, 也有两种. 本文介绍 Windows 版存储网关的实现方法和基于 Linux 下 fuse 的存储网关实现方法相似.

其运行机制如下图 2(图中虚框部分为代码需要实现的部分, 实框部分为 Windows 操作系统已提供的功能部分; RDFL 为 Radiocom DFS 文件系统过滤驱动模块; dfsclient 为用户态分布式文件系统存储网络应用程序).

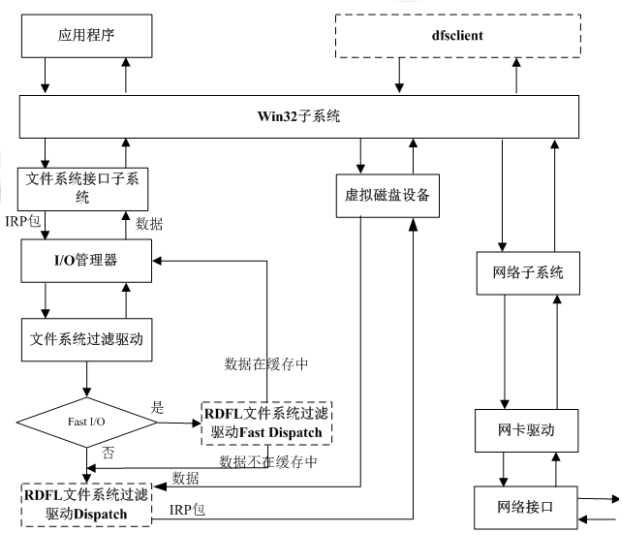

图 2 Windows 存储网关运行机制示意图

Windows 版存储网关实现流程: Windows 系统应用程序通过 Win32 子系统调用文件系统接口子系统接口, 文件系统接口子系统通过 I/O 管理器将调用请求传递给 RDFL, RDFL 将接收到的请求由虚拟磁盘设备将请求转发给 dfsclient 存储网关程序, dfsclient 通过网络连接与分布式文件系统集群进行数据交互: 将数据写到分布式文件系统或者从分布式文件系统获取数据, 最后, 将写入数据状态或者获取数据内容返回给应用程序.

Windows 版存储网关代码由三部分组成: (1) RDFL 文件系统过滤驱动模块部分, 该部分驱动模块包含两个功能, RDFL 驱动模块的加载和文件系统对象及虚拟磁盘设备对象的创建. RDFL 驱动模块的加载在 Windows 启动时由后台服务自动加载, 文件系统对象和虚拟磁盘设备对象在 dfsclient 启动时创建, 结束时删除. (2) RDFL 文件系统过滤驱动的用户态适配部分, 该部分的主要功能是将 RDFL 驱动模块需要实现的回调函数映射到用户态, 由 dfsclient 实现一组与

RDFL 驱动回调函数对应的回调函数集合; 同时还有创建, 删除, 管理文件系统对象和虚拟磁盘对象的接口. (3) dfsclient 分布式文件系统存储网关部分, 该部分主要功能是将对分布式文件系统的操作原语转化成 RDFL 提供的一组回调函数实现.

2.1 RDFL 文件系统驱动

RDFL 文件系统驱动在操作系统启动时候, 由后台服务进程 RDFSMounter 自动将之加载到操作系统内核中, 当 RDFL 文件系统驱动被加载到内核中时, RDFL 会创建一个名称为 RDFLCtl 的设备对象, 用户态可以通过该设备对象设置和查询 RDFL 驱动的状态, 以及创建和删除文件系统对象和虚拟磁盘设备对象.

RDFL 文件系统驱动的注册可以分为三步: (1) 通过 IoCreateDeviceSecure 创建名称为 RDFLCtl 的虚拟磁盘设备, 建立起用户态和内核态之间通信的通道; (2) 将回调函数关联到 DRIVER_OBJECT 结构对应对象 DriverObject 中字段 MajorFunction 的回调函数集合中, 这些注册的回调函数包括处理文件系统请求的 Create、Read、Write 和 Close 函数, 也包括处理查询、设置、增加和删除虚拟磁盘设备对象命令的事件处理函数. (3) 通过 FsRtlRegisterFileSystemFilterCa-backs 函数将 RDFL 文件系统驱动注册到 Windows 操作系统中. 通过以上三步即可完成 RDFL 文件系统驱动模块的注册.

dfsclient 程序启动的时候, 首先会通过网络连接查询分布式文件系统集群是否存在. 在存在的条件下, dfsclient 调用 RDFL 用户态提供的主接口函数, 触发内核创建文件系统对象和虚拟磁盘对象: 内核通过 IoCreateDeviceSecure 函数创建创建虚拟磁盘对象, 然后在创建一个文件系统对象, 将虚拟磁盘对象和文件系统对象关联起来. 随后, 内核创建一个内核线程监听来自网络虚拟磁盘的各种事件消息, 当接收到事件的时候, 对事件进行处理.

2.2 RDFL 用户态动态库

RDFL 内核驱动的用户态部分, 对内核态 RDFL 驱动程序提供的功能进行了封装, 提供了一组管理函数. 这些函数包括三部分: (1) 文件系统对象和虚拟磁盘对象的创建、删除、查询和设置, 其中创建函数在 dfsclient 启动的时候被调用, 删除函数在 dfsclient 函数结束的时候被调用, 查询和设置函数被 RDFSMounter 调用, 用以对文件系统对象和虚拟磁盘对象状态进行查询和设置. (2) 文件系统回调函数 MajorFunction 到用户态的映射函数, 这些回调函数包括 CreateFile、OpenDirectory、CreateDirectory、ReadFile、WriteFile、CloseFile 等函数, dfsclient 存储网关的主要功能就是将这些回调函数转换成对分布式文件系统的操作. (3) 最后一些包括调试开关的打开关闭, 版本信息的查询等函数.

2.3 dfsclient 存储网关

从程序设计上而言, dfsclient 包含三部分: (1) 调度转发子系统通过分布式文件系统操作原语实现 RDFL.dll 中要求实现的类似 Windows 文件系统标准接口的回调接口, 调度转发子系统也包括了多线程数据并发读写, 读写数据缓存, 元数据缓存等功能. (2) RDFL.dll 模块接收来自 RDFL.sys 的对文件进行操作的事件, 并将事件转发给调度转发子系统, 由调度转发子系统 “翻译”成分布式文件系统操作原语, 实现对分布式文件系统的操作. (3) 分布式文件系统操作原语是由分布式文件系统提供的, 基于 socket 的可以对文件及文件夹进行操作的函数集合.

其设计框架如下图 3.

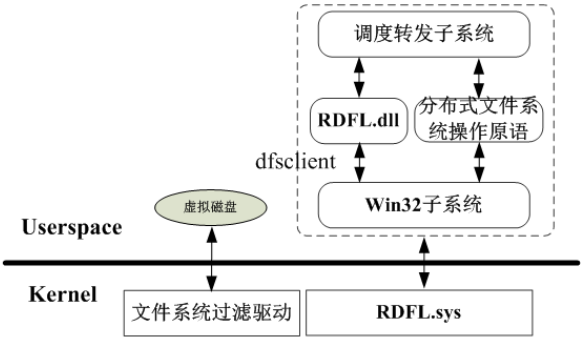

图 3　dfsclient 设计框架示意图

虚拟磁盘对文件的操作, 经过 RDFL(RDFL.sys 和 RDFL.dll)模块以后, 有些对文件的操作, RDFL 模块会将一个对虚拟磁盘的操作分解成多个对 Windows 存储网关的回调操作. 特别是读写操作, 在支持多线程的情况下, 会被分解成多个操作且可能并发执行, 甚至出现文件关闭操作早于读写操作被调用的情况, 这时候需要注意多线程操作的同步问题. 对于 dfsclient 存储网关而言, 一些回调函数会对应多个对分布式文件系统服务器的操作, 特别是对文件内容修改的操作, 有时候还会导致 Linux 集群内部服务器之间进行多次操作.

大多数情况, 客户端都会使用多线程机制, 以便更好的利用网络性能, Windows 版存储网关可以使用线程开发库(pthreads-win32[3]), 该线程库是 Windows 下的 pthread 库, 能够很好的兼容 Linux 操作系统, 减少开发过程中的移植性问题.

2.4 Windows 系统存储网关和 Dokan、Samba 间差异

RDFL 和 Dokan 对比: Dokan 可以帮助程序员在 windows 系统下轻松建立用户级文件系统, 不需要写设备驱动, 其与 FUSE(Linux user mode file system)类似. RDFL 和 Dokan 具有相似的功能, 都可以帮助用户建立用户级文件系统.

从设计初衷上看, Dokan 基本上是按照 Windows 文件系统接口设计的, 更看重的是对 Windows 文件系统接口的兼容. RDFL 基本上是按照 Linux 文件系统接口设计的, 对于在 Linux 上已经实现了 fuse 客户端的应用而言, 从 Linux 到 Windows 上的迁移将更加方便. 特别, 对文件名的大小写识别, 文件名的字符集转换, 文件数据的并发读取和写入等问题可以更容易的解决. 从调用流程上, 当调用 GetFileAttributes, SetAllocationSize, DeleteFile 的时候, Dokan 会进行 CreateFile, Other Functions, Cleanup, CloseFile 等一个流程的函数调用, 这样的流程设计必然降低了 Other Functions 的响应速度. RDFL 只将逻辑上必要 ReadFile, WriteFile 等操作按照 CreateFile, Other Functions, CloseFile 的流程进行调用. 对于不需要 CreateFile 的将直接调用对应的函数. 从缓存内存上, RDFL 在内核态和用户态提供了一套缓存系统, 内核态缓存功能在程序启动的时候可以配置到内核, 用户态缓存功能, 用户通过接口调用, 可以对元数据, 内容数据进行缓存管理. 缓存功能包括设定缓存数据的量, 缓存数据的老化时间, 缓存的添加, 删除等. 该功能和 fuse 提供的缓存功能相似, Dokan 没有提供此种功能.

Windows 系统存储网关和 Samba 对比: Samba 是在 Linux 和 UNIX 系统上实现 SMB 协议的一个免费软件, 由服务器及客户端程序构成. Samba 的主要目就是要用来沟通 Windows 与 Unix Like 这两个不同的作业平台.

从客户使用的角度来看, Windows 系统存储网关和 Samba 客户端没有差别, 都实现了文件的创建, 删除, 读写访问. 从协议上看, Samba 通过 SMB 协议进行客户端和服务器之间的交互; Windows 系统存储网关使用 HDFS 提供的协议进行交互, SMB 协议是在 NetBIOS 协议的基础上实现的, HDFS 是在 RPC 协议的基础上实现的. 从实现上看, Samba 的 Windows 客户端是内置在 Windows 系统中的; Windows 系统存储网关是基于 Windows 文件过滤驱动 RDFL 实现的用户态应用程序, 具有更好的管理性和扩展性. 能够提供更好的用户名、密码认证和文档管理机制, 可以通过 HDFS 数据通信协议更好的扩展数据管理功能. 从服务器来看, Samba 服务器共享本机提供的存储服务; HDFS 集群将文件的元数据和内容数据分离, Windows 客户端从元数据服务器获得文件的元数据信息, 从存储数据服务器获得文件的内容数据信息, 这种分布式的实现方式, 不但可以提高数据可靠性和可用性, 也提高了多客户端数据获取的吞吐量.

3 Windows系统存储网关性能测试

以下是基于 HDFS 分布式文件系统的 Windows 版存储网关和 Linux 版存储网关的读写性能对比, 读写数据为 4G 大小的大文件, 服务器配置如下:

元数据服务器(NameNode)配置: 一块千兆网卡, 8G 内存, 4 个 2 核 CPU.

存储数据服务器(DataNode)配置: 两块千兆网卡(bonding), 15000 转 8M 缓存磁盘, 4 磁盘 RAID0, 8G 内存, 4 个 2 核 CPU.

存储网络客户端配置: 一块千兆网卡, 15000 转 8M 缓存磁盘, 4 磁盘 RAID0, 8G 内存, 4 个 2 核 CPU.

通过两层千兆交换机将元数据服务器, 存储数据服务器和 Windows, Linux 版存储网关连接起来. 数据存储采用一个副本方式存储, 存在三台存储数据服务器.

对比 Windows 存储网关和 Linux 存储网关数据转发的性能如图 4 所示.

其性能测试结果如下:

整体上, Linux 下使用 fuse 的存储网关比 Windows 下使用 RDFL 的存储网关性能都要高, 特别是在单次读写的数据流量上——经过分析这是由于 Windows 操作系统自身系统策略决定的, WindowsXP 操作系统每 15 毫秒主动进行一次系统调度, 不管该任务优先级高低. Windows 整体写的效率会比读数据的效率略微高点是由于 Windows 存储网关写操作使用多线程并发发送数据导致的. 在两次并发读写上, Linux 版存储网关和 Windows 版存储网关性能基本相同. 在客户端或存

储数据服务器磁盘性能不足的情况下, 读写并发数量的增加会导致性能的降低.

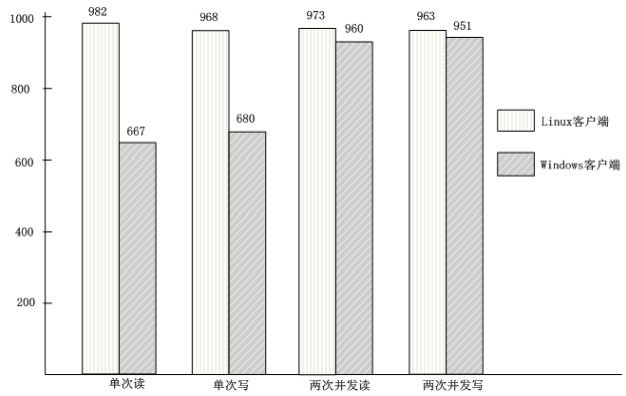

图 4 Windows、Linux 版存储网关性能对比示意图

4 总结

使用 Windows 文件系统过滤驱动开发类似于 Linux 下 fuse 的驱动程序 RDFL, 可以比较容易的开发 Windows 版存储网关, 极大的缩短开发周期. 当然, 在开发过程中存在很多问题, 主要是 Windows 操作系统和 Linux 操作系统之间的兼容性问题, 但是, 这些问题对数据访问的影响基本可以忽略. 对基于 RDFL 开发存储网关而言, pthread_win32 可以使从 Linux 到 Windows 下的移植变得容易简单, 特别对 Windows 多线程接口不熟悉的开发者.

参考文献

1 The Google File System.

2 黄华.蓝鲸分布式文件系统资源管理[学位论文].北京:中国科学院研究生院, 2005.

3 ftp://sourceware.org/pub/pthreads-win32/.

4 http://dokan-dev.net/en/download/.