可行性报告

## FUSE

可用参考资料：

<https://www.ibm.com/developerworks/cn/linux/l-fuse/#resources>

<https://github.com/libfuse/libfuse>

FUSE (Filesystem in Userspace) is an interface for userspace programs to export a filesystem to the Linux kernel. The FUSE project consists of two components: the fuse kernel module (maintained in the regular kernel repositories) and the libfuse userspace library (maintained in this repository). libfuse provides the reference implementation for communicating with the FUSE kernel module.

A FUSE file system is typically implemented as a standalone application that links with libfuse. libfuse provides functions to mount the file system, unmount it, read requests from the kernel, and send responses back. libfuse offers two APIs: a "high-level", synchronous API, and a "low-level" asynchronous API. In both cases, incoming requests from the kernel are passed to the main program using callbacks. When using the high-level API, the callbacks may work with file names and paths instead of inodes, and processing of a request finishes when the callback function returns. When using the low-level API, the callbacks must work with inodes and responses must be sent explicitly using a separate set of API functions.

文件系统是一种用来存储和组织计算机文件、目录及其包含的数据的方法，它使文件、目录以及数据的查找和访问得到简化。文件系统能提供丰富的扩展能力。它可以编写成底层文件系统的一个封装程序，从而对其中的数据进行管理，并提供一个增强的、具有丰富特性的文件系统（例如 cvsfs-fuse，它为 CVS 提供了一个文件系统的接口；或 Wayback 文件系统，它提供了一种用于保留原始数据文件的文件备份机制）。

FUSE 是Linux下同于支持用户空间文件系统(Filesystem in Userspace)的内核模块，可以开发功能完备的，在用户态实现的文件系统：其具有简单的 API 库，无需进行内核编程，可以被非特权用户访问，并可以安全的实施。更重要的是，FUSE 以往的表现充分证明了其稳定性。

就文件系统来说，用户空间的文件系统就不再是新奇的设计了。用户空间文件系统的商业实现与学术实现的实例包括：

1）LUFS 是一个混合用户空间的文件系统框架，它对用于任何应用程序无数的文件系统提供透明支持。大部分LUFS 包括一个内核模块和一个用户空间的守护进程。从根本上来说，它将大部分 VFS 调用都委托给一个专用的守护进程来处理。

2）UserFS 让用户进程可以像普通的文件系统一样进行加载。这种概念性的原型提供了 ftpfs，这可以使用文件系统接口提供匿名 FTP 访问。

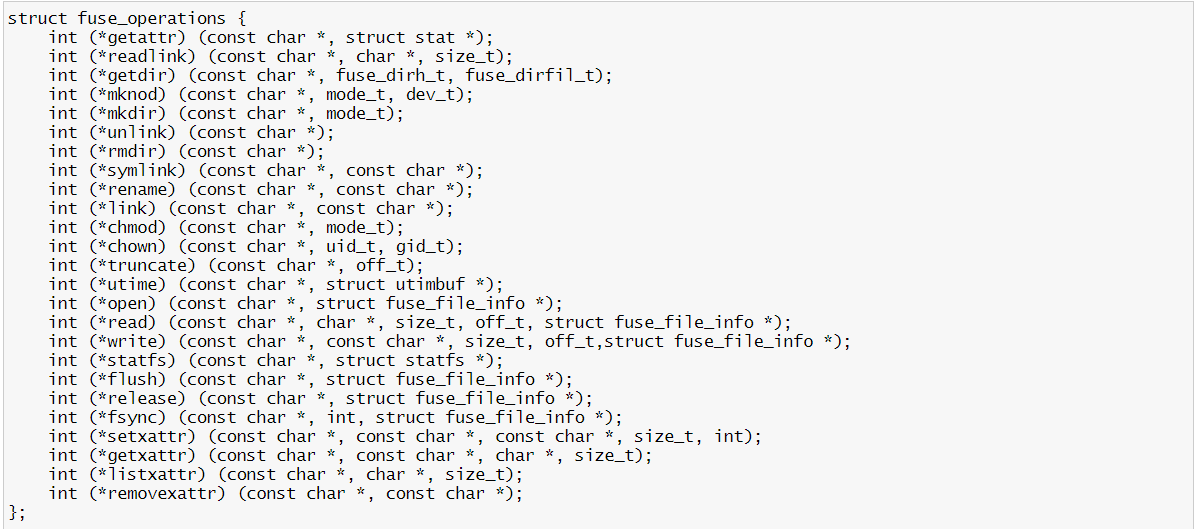
3）Ufo Project 是为 Solaris 提供的一个全局文件系统，它允许用户将远程文件真正当作本地文件一样对待。

4）OpenAFS 是 Andrew FileSystem 的一个开源版本。

5）CIFS 是 Common Internet FileSystem 的简称。

与这些商业实现和学术实现不同，FUSE 将这种文件系统的设计能力带到了 Linux 中来。由于 FUSE 使用的是可执行程序（而不像 LUFS 一样使用的是共享对象），因此可以简化程序的调试和开发。FUSE 可以在 2.4.x 和 2.6.x 的内核上使用，除C和C++外，也可以支持 Java绑定。

要使用 FUSE 来创建一个文件系统，需要声明一个fuse\_operations类型的结构变量，并将其传递给fuse\_main函数。fuse\_operations结构中有一个指针，指向在执行适当操作时需要调用的函数。下为Fuse\_operation结构和其中需要的函数



## Storj

可用参考资料：

<https://storj.io/storj.pdf>

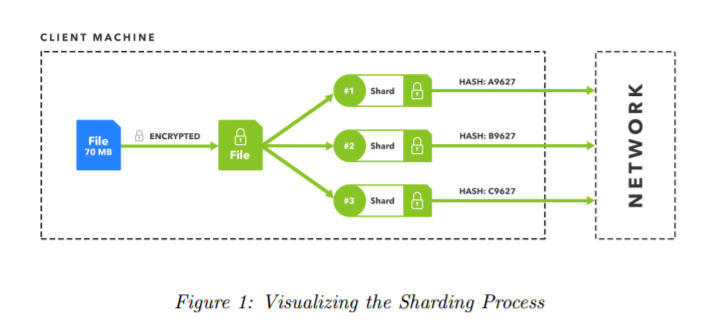
<https://storj.io/>

<https://github.com/Storj/>

Storj是基于blockchain技术和点对点(peer to peer)协议的开源、去中心化存储平台，提供最安全，私有和加密的云存储服务。Storj打算使用多重区块链存储元数据，作为记录文件存储位置和冗余度的目录。一旦区块链变得特别庞大，他们将使用Merkle树加速进程，类似于比特币SPV钱包工作原理。

实行客户端加密的点对点云存储网络可以使用户不依赖第三方存储供应方来传输和共享数据。去中心存储可以减少大多数传统存储带来的数据错误与断供，并显著提高数据安全、隐私和数据可靠性。Storj可用于网络上的点间协同传输数据、检验数据的完整性和可用性，恢复数据。支持端对端加密(end-to-end encryption)，采用点对点传输技术(peer to peer)，快速高效，并具有高度可用性。

切分(shard)是加密文件在分布式网路上存储的分配，在安全、隐私、性能和可用性方面有重要作用。文件在被切分前必须先经过加密。这种策略可以保护数据内容不被存储提供方看到，以保护隐私。只有数据拥有者持有对密钥的完整控制，知道文件的切分方式和每个切分的存储位置，进而可以查看和使用数据。当网络中的切分变多，如果不提前知道切分的位置，会使定位给定切分集变得格外困难，这就意味着文件的安全性和网络大小的平方是成比例的。



The sharding process

1. Files are encrypted.
2. Encrypted files are split into shards, or multiple files are combined to form a shard.
3. Audit pre-processing is performed for each shard (see Section 2.3). 4. Shards may be transmitted to the network.

## Sync: Rsync

可用参考资料：

<http://www.linuxidc.com/Linux/2016-10/136143.htm>

<http://man.linuxde.net/rsync>

1. sync命令是一个远程数据同步工具，可通过LAN/WAN快速同步多台主机间的文件。rsync全称remote sync，是一种更高效、可以本地或远程同步的命令，之所以高效是因为rsync使用所谓的“rsync算法”，会对需要同步的源和目的进度行对比，只同步有改变的部分，而不是每次都整份传送，因此速度相当快，比scp命令高效得多。但是rsync本身是一种非加密的传输，可以借助-e选项来设置具备加密功能的承载工具进行加密传输。
2. rsync的工作模式：
3. shell模式，也称作本地模式
4. 远程shell模式，此时可以利用ssh协议承载其数据传输过程
5. 列表模式，其工作方式与ls相似，仅列出源的内容：-nv
6. 服务器模式，此时，rsync可以工作在守护进程，能够接收客户端的数据请求；在使用时，可以在客户端使用rsync命令把文件发送到守护进程，也可以像服务器请求获取文件
7. rsync命令选项

-v, --verbose 详细模式输出。

-q, --quiet 精简输出模式。

-c, --checksum 打开校验开关，强制对文件传输进行校验。

-a, --archive 归档模式，表示以递归方式传输文件，并保持所有文件属性，等于-rlptgoD。 -r, --recursive 对子目录以递归模式处理。

-R, --relative 使用相对路径信息。

-b, --backup 创建备份，也就是对于目的已经存在有同样的文件名时，将老的文件重新命名为~filename。可以使用--suffix选项来指定不同的备份文件前缀。

--backup-dir 将备份文件(如~filename)存放在在目录下。

-suffix=SUFFIX 定义备份文件前缀。

-u, --update 仅仅进行更新，也就是跳过所有已经存在于DST，并且文件时间晚于要备份的文件，不覆盖更新的文件。

-l, --links 保留软链结。

-L, --copy-links 想对待常规文件一样处理软链结。

--copy-unsafe-links 仅仅拷贝指向SRC路径目录树以外的链结。

--safe-links 忽略指向SRC路径目录树以外的链结。

-H, --hard-links 保留硬链结。

-p, --perms 保持文件权限。

-o, --owner 保持文件属主信息。

-g, --group 保持文件属组信息。

-D, --devices 保持设备文件信息。

-t, --times 保持文件时间信息。

-S, --sparse 对稀疏文件进行特殊处理以节省DST的空间。

-n, --dry-run现实哪些文件将被传输。

-w, --whole-file 拷贝文件，不进行增量检测。

-x, --one-file-system 不要跨越文件系统边界。

-B, --block-size=SIZE 检验算法使用的块尺寸，默认是700字节。

-e, --rsh=command 指定使用rsh、ssh方式进行数据同步。

--rsync-path=PATH 指定远程服务器上的rsync命令所在路径信息。

-C, --cvs-exclude 使用和CVS一样的方法自动忽略文件，用来排除那些不希望传输的文件。

--existing 仅仅更新那些已经存在于DST的文件，而不备份那些新创建的文件。

--delete 删除那些DST中SRC没有的文件。

--delete-excluded 同样删除接收端那些被该选项指定排除的文件。

--delete-after 传输结束以后再删除。

--ignore-errors 及时出现IO错误也进行删除。

--max-delete=NUM 最多删除NUM个文件。

--partial 保留那些因故没有完全传输的文件，以是加快随后的再次传输。

--force 强制删除目录，即使不为空。

--numeric-ids 不将数字的用户和组id匹配为用户名和组名。

--timeout=time ip超时时间，单位为秒。

-I, --ignore-times 不跳过那些有同样的时间和长度的文件。

--size-only 当决定是否要备份文件时，仅仅察看文件大小而不考虑文件时间。

--modify-window=NUM 决定文件是否时间相同时使用的时间戳窗口，默认为0。

-T --temp-dir=DIR 在DIR中创建临时文件。

--compare-dest=DIR 同样比较DIR中的文件来决定是否需要备份。

-P 等同于 --partial。

--progress 显示备份过程。

-z, --compress 对备份的文件在传输时进行压缩处理。

--exclude=PATTERN 指定排除不需要传输的文件模式。

--include=PATTERN 指定不排除而需要传输的文件模式。

--exclude-from=FILE 排除FILE中指定模式的文件。

--include-from=FILE 不排除FILE指定模式匹配的文件。

--version 打印版本信息。

--address 绑定到特定的地址。

--config=FILE 指定其他的配置文件，不使用默认的rsyncd.conf文件。

--port=PORT 指定其他的rsync服务端口。

--blocking-io 对远程shell使用阻塞IO。

-stats 给出某些文件的传输状态。

--progress 在传输时现实传输过程。

--log-format=formAT 指定日志文件格式。

--password-file=FILE 从FILE中得到密码。

--bwlimit=KBPS 限制I/O带宽，KBytes per second。

-h, --help 显示帮助信息。

## SSL

## Encryption: AES

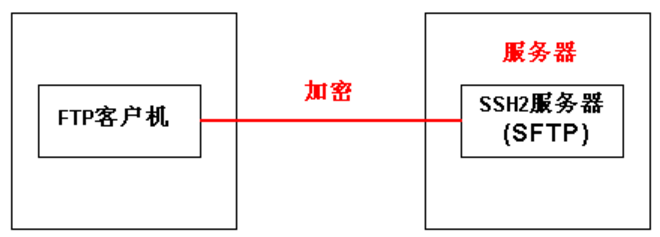
## Server: SFTP

1. SFTP，即安全文件传送协议(Secure File Transfer Protocol)，是一个交互式文件传输程式，可以为传输文件提供一种安全的加密方法。SFTP与FTP有着几乎一样的语法和功能，但它进行加密传输，比FTP有更高的安全性。SFTP为SSH的一部分，是一种传输档案至 Blogger 伺服器的安全方式。其实在SSH软件包中，已经包含了一个叫作SFTP(Secure File Transfer Protocol)的安全文件传输子系统，SFTP本身没有单独的守护进程，它必须使用sshd守护进程（端口号默认是22）来完成相应的连接操作，所以从某种意义上来说，SFTP并不像一个服务器程序，而更像是一个客户端程序。SFTP同样是使用加密传输认证信息和传输的数据，所以使用SFTP是非常安全的。但是，由于这种传输方式使用了加密/解密技术，所以传输效率比普通的FTP要低得多。
2. 常用登陆方式：

格式：sftp <user>@<host>

通过sftp连接<host>，端口为默认的22，指定用户<user>。

1. 工作模式



SFTP的工作模式，它是作为SSH2的一个子服务工作的。

3.基本使用

(1)文件下载

get [-Ppr] remote [local]

如：get test.cpp ./Project/

将远程当前目录下的文件test.cpp下载到本地当前目录的Project文件夹中。

(2)文件上传

put [-Ppr] local [remote]

如：put /home/liu/Software/RHEL\_5.5\ x86\_64.iso /home/xudong/Blog/

将本地/home/liu/Software/目录下的ios文件传送到远程登陆主机的/home/xudong/Blog/目录下。

(3)其他命令

-> Help：建立连接后，查看sftp支持的命令选项。

-> pwd和lpwd：pwd是看远端服务器的目录，即sftp服务器默认的当前目录。

lpwd是看linux本地目录。

-> ls和lls：ls查看 sftp服务器下当前目录下的文件，lls查看linux当前目录下的文件。

-> ![command]：在linux上执行command所列命令。例如!ls是列举linux当前目录下的文件， !rm a.txt是删除linux当前目录下的a.txt文件。

-> exit和quit：退出。

## Cloud disk: proto