```
ls *.tar.gz | xargs -n1 tar xzvf
# gzip -d *.tar.gz

-------------------------------------------------------------------------------------------------------------------------------
--------------------
#               Installing Perl Module Dependencies                #
-------------------------------------------------------------------------------------------------------------------------------
--------------------
service vmware-tools start
cd /mnt/hgfs/www/MogileFS.rpm/

rpm -ivh /mnt/hgfs/share/perl-IO-stringy-2.110-1.2.el5.rf.noarch.rpm
rpm -ivh /mnt/hgfs/share/perl-HTML-Tagset-3.20-1.el5.rf.noarch.rpm
rpm -ivh /mnt/hgfs/share/perl-HTML-Parser-3.56-1.el5.rf.i386.rpm
rpm -ivh /mnt/hgfs/share/libghttp-1.0.9-10.99_2.0.el5.i386.rpm
rpm -ivh /mnt/hgfs/share/perl-HTTP-GHTTP-1.07-1.el5.rf.i386.rpm
rpm -ivh /mnt/hgfs/share/perl-libwww-perl-5.803-2_6.0.el5.noarch.rpm
rpm -ivh /mnt/hgfs/share/perl-MogileFS-Client-1.08-1.fc8.noarch.rpm

rpm -ivh /mnt/hgfs/share/perl-Compress-Raw-Zlib-2.008-1.el5.rf.i386.rpm
rpm -ivh /mnt/hgfs/share/perl-IO-Compress-Base-2.008-1.el5.rf.noarch.rpm
rpm -ivh /mnt/hgfs/share/perl-IO-Compress-Zlib-2.008-1.el5.rf.noarch.rpm
rpm -ivh /mnt/hgfs/share/perl-Compress-Zlib-2.008-1.el5.rf.noarch.rpm
rpm -ivh /mnt/hgfs/share/perl-MogileFS-Utils-2.12-1.el5.noarch.rpm

rpm -ivh /mnt/hgfs/share/perl-Net-Netmask-1.9015-1.el5.rf.noarch.rpm
rpm -ivh /mnt/hgfs/share/perl-Net-Daemon-0.43-1.el5.rf.noarch.rpm
rpm -ivh /mnt/hgfs/share/perl-PlRPC-0.2020-1.el5.rf.noarch.rpm
rpm -ivh /mnt/hgfs/share/perl-DBI-1.602-1.el5.rf.i386.rpm
rpm -ivh /mnt/hgfs/share/mysqlclient15-5.0.45-1.el5.remi.i386.rpm
rpm -ivh /mnt/hgfs/share/perl-DBD-mysql-4.006-1.el5.rf.i386.rpm
rpm -ivh /mnt/hgfs/share/perl-Gearman-1.09-1.el5.noarch.rpm
rpm -ivh /mnt/hgfs/share/perl-Sys-Syscall-0.22-1.el5.rf.noarch.rpm
rpm -ivh /mnt/hgfs/share/perl-Danga-Socket-1.58-1.el5.rf.noarch.rpm
rpm -ivh /mnt/hgfs/share/perl-Gearman-Client-Async-0.94-3.el5.noarch.rpm
rpm -ivh /mnt/hgfs/share/perl-Gearman-Server-1.09-1.el5.noarch.rpm
rpm -ivh /mnt/hgfs/share/perl-BSD-Resource-1.2901-1.el5.rf.i386.rpm
rpm -ivh /mnt/hgfs/share/perl-IO-AIO-2.51-1.el5.rf.i386.rpm
rpm -ivh /mnt/hgfs/share/Perlbal-1.59-1.el5.noarch.rpm

rpm -i /mnt/hgfs/share/perl-mogilefs-server-2.20-4.el5.src.rpm
cd /usr/src/redhat/SPECS
rpmbuild -bp perl-mogilefs-server.spec
```

```
cd /usr/src/redhat/BUILD/mogilefs-server-2.20/
perl Makefile.PL

make
make install
cd ../..
rm -rf BUILD/mogilefs-server-2.20
rm -rf SPECS/perl-mogilefs-server.spec
rm -rf SOURCES/mog*

 vi /root/.bash_profile
#PATH=$PATH:$HOME/bin
PATH=$PATH:$HOME/bin:/usr/local/mysql/bin
su -

# 在VMware上做实验,如果make编译的时候出现, 系统时间报错,则执行如下,再次make;
find . -type f -exec touch {} \;

--------------------------------------------------------------------------------------------------------------------------------
#              Creating a Database                #
--------------------------------------------------------------------------------------------------------------------------------
mysql -p
CREATE DATABASE mogilefs_fk;
GRANT ALL ON mogilefs_fk.* TO 'mogile_fk'@'%';
# grant all privileges on *.* to 'mogile_fk'@'%';
SET PASSWORD FOR 'mogile_fk'@'%' = OLD_PASSWORD('mogile_pw');
FLUSH PRIVILEGES;
quit

mogdbsetup --dbhost=192.168.1.20 --dbname=mogilefs_fk --dbuser=mogile_fk --dbpass=mogile_pw
y

mkdir -p /etc/mogilefs
mkdir -p /var/mogdata

# Tracker Configuration
vi /etc/mogilefs/mogilefsd.conf
daemonize = 1
db_dsn = DBI:mysql:mogilefs_fk:192.168.1.20
db_user = mogile_fk
db_pass = mogile_pw
```

```
listen = 0.0.0.0:6001
conf_port = 6001
query_jobs = 2
listener_jobs = 10
delete_jobs = 1
replicate_jobs = 5
reaper_jobs = 1
# mog_root = /var/mogdata       \\ 原先老版本使用NFS,当前版本已经不再使用,可以忽略

# mogilefsd  与 syslog 有依存关系, 启动syslog服务
chkconfig syslog on
service syslog start

# 若 syslog 造成 mogilefsd 意外关闭, 尝试以下修改 "或许" 能解决
vi /usr/lib/perl5/site_perl/5.8.8/MogileFS/Server.pm
    # 可选, 或许会造成 mogilefsd 无法启动
    setlogsock('udp');
    # Sys::Syslog::openlog('mogilefsd', 'pid', 'daemon');
    Sys::Syslog::openlog('mogilefsd', 'pid,nofatal', 'daemon');

adduser mogile_fk

vi /etc/init.d/mogilefsd
# ==========/etc/init.d/mogilefsd==============
#!/bin/bash
#
# mogilefsd    Startup script for the MogileFS tracker
#
# chkconfig: - 85 15
# description: MogileFS tracker
# processname: mogilefsd
# config: /etc/mogilefs/mogilefsd.conf
# pidfile: /var/run/mogilefsd.pid

# Source function library.
. /etc/rc.d/init.d/functions

# Path to the apachectl script, server binary, and short-form for messages.
lockfile=${LOCKFILE-/var/lock/mogilefsd}
RETVAL=0

dbUser=${mogile_fk}
```

```
start() {
        echo -n $"Starting mogilefsd: "
        sudo -u ${dbUser} /usr/bin/mogilefsd -c /etc/mogilefs/mogilefsd.conf --daemon
        RETVAL=$?
        echo
        [ $RETVAL = 0 ] && touch ${lockfile}
        return $RETVAL
}
stop() {
        echo -n $"Stopping $prog: "
        killproc mogilefsd
        RETVAL=$?
        echo
        [ $RETVAL = 0 ] && rm -f ${lockfile}
}
reload() {
    echo -n $"Reloading mogilefsd: "
    killproc mogilefsd -HUP
    RETVAL=$?
    echo
}

# See how we were called.
case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  status)
        status mogilefsd
        RETVAL=$?
        ;;
  restart)
        stop
        start
        ;;
  reload)
        reload
        ;;
  *)
        echo $"Usage: mogilefsd {start|stop|restart|reload|status}"
```

```
        exit 1
esac

exit $RETVAL
# ==============EOF=====================
chmod 755 /etc/rc.d/init.d/mogilefsd
chkconfig --add mogilefsd
chkconfig mogilefsd on
service mogilefsd start
# sudo -u mogile_fk /usr/bin/mogilefsd -c /etc/mogilefs/mogilefsd.conf --daemon
ps -ef |grep mogilefsd

# 取消执行sudo命令时需要终端
vi /etc/sudoers
#Defaults    requiretty

# Storage Server Configuration
vi /etc/mogilefs/mogstored.conf
daemonize = 1
maxconns = 10000
httplisten = 0.0.0.0:7500
mgmtlisten = 0.0.0.0:7501
docroot = /var/mogdata

vi /etc/mogilefs/mogilefs.conf
trackers = 192.168.1.20:6001, 192.168.1.21:6001

vi /etc/mogilefs/mogtool.conf
trackers = 192.168.1.20:6001, 192.168.1.21:6001
domain = fkootestdomain
class = fkootestdata
# lib = /usr/lib/perl5/5.8.8/
lib = /usr/lib/perl5/vendor_perl/5.8.8/
overwrite = 1
chunksize = 32M
receipt = admin@fkoo.com
verify = 1
concurrent = 3

mogadm host add fkoomogilestorage1 --ip=192.168.1.20 --port=7500 --status=alive
mogadm host add fkoomogilestorage2 --ip=192.168.1.21 --port=7500 --status=alive

mogadm host list
```

```
mogadm device add fkoomogilestorage1 1
mogadm device add fkoomogilestorage2 2

mogadm device list

mkdir -p /var/mogdata/dev1
mkdir -p /var/mogdata/dev2

vi /etc/init.d/mogstored
# ==========/etc/init.d/mogstored==============
#!/bin/bash
#
# mogstored  - Startup script for the MogileFS storage node
#
# chkconfig: - 85 15
# description: MogileFS storage node
# processname: mogstored
# config: /etc/mogilefs/mogstored.conf
# pidfile: /var/run/mogstored.pid

# Source function library.
. /etc/rc.d/init.d/functions

# Path to the apachectl script, server binary, and short-form for messages.
lockfile=${LOCKFILE-/var/lock/mogstored}
RETVAL=0

start() {
         echo -n $"Starting mogstored: "
         /usr/bin/mogstored --config /etc/mogilefs/mogstored.conf --daemonize > /dev/null
         RETVAL=$?
         echo
         [ $RETVAL = 0 ] && touch ${lockfile}
         return $RETVAL
}
stop() {
         echo -n $"Stopping $prog: "
         killproc mogstored
         RETVAL=$?
         echo
         [ $RETVAL = 0 ] && rm -f ${lockfile}
}
```

```
reload() {
     echo -n $"Reloading mogstored: "
     killproc mogstored -HUP
     RETVAL=$?
     echo
}

# See how we were called.
case "$1" in
   start)
         start
         ;;
   stop)
         stop
         ;;
   status)
         status mogstored
         RETVAL=$?
         ;;
   restart)
         stop
         start
         ;;
   reload)
         reload
         ;;
   *)
         echo $"Usage: mogstored {start|stop|restart|reload|status}"
         exit 1
esac

exit $RETVAL
# ==============EOF=====================
chmod 755 /etc/rc.d/init.d/mogstored
chkconfig --add mogstored
chkconfig mogstored on
service mogstored start
# mogstored --daemon
ps -ef | grep mogstored

mogadm check

mogadm domain add fkootestdomain
```

```
mogadm class add fkootestdomain fkootestclass

# Use modtool for adding and retriving the data.
mogtool inject file-name key-name
mogtool extract key-name file-name

# for large files, >64M
mogtool inject --bigfile file-name key-name
mogtool extract --bigfile key-name file-name

# for directories
mogtool inject --bigfile dir-name key-name
mogtool extract --bigfile dir-name file-name

vi test.pl
#============test.pl====================
use MogileFS::Client;
my $mogfs = MogileFS::Client->new(domain=>'fkootestdomain', hosts=>['192.168.1.20:6001'], root=>'/var/mogdata',);
my $fh = $mogfs->new_file("file_key", "fkootestclass");
die $fh unless $fh->print($mogfs->readonly);
my $content = "file.txt";
@num = $mogfs->store_content("file_key","fkootestclass",$content);
print "@num \n";
my $file_contents = $mogfs->get_file_data("file_key");
print "$file_contents \n";
#$mogfs->delete("file_key");
$fh->print($file_contents);
@urls = $mogfs->get_paths("file_key");
print "@urls \n";
#============EOF=======================
[root@R20_51 ~]# perl test.pl
8
SCALAR(0x8e68b74)
http://192.168.1.20:7500/dev1/0/000/000/0000000014.fid

vi dbtest.pl
#============dbtest.pl==================
#!/usr/bin/perl
# DBI is perl module used to connect to the database
use DBI;

# hostname or ip of server (for local testing, localhost should work)
$config{'dbServer'} = "192.168.1.20";
```

```
$config{'dbUser'} = "mogile_fk";
$config{'dbPass'} = "mogile_pw";
$config{'dbName'} = "mogilefs_fk";
$config{'dataSource'} = "DBI:mysql:$config{'dbName'}:$config{'dbServer'}";

# Connect to MySQL
my $dbh = DBI->connect($config{'dataSource'},$config{'dbUser'},$config{'dbPass'}) or
die "Can't connect to $config{'dataSource'}<br>$DBI::errstr";
 print "Connected successfully<br>";
$dbh->disconnect();
#==============EOF======================
[root@R20_51 ~]# perl dbtest.pl
Connected successfully<br>
```

```
# 确认rc.local是否启动命令, 需要reboot后观察终端显示
vi /etc/rc.d/rc.local
echo "Press any key to continue...."
read AnyKey
```

---

介绍
首先可以去看官方的wiki：http://mogilefs.pbwiki.com/。（有可能被GFW了，那么你就安装一个Firefox的gladder插件就可以看了）。
偶简单记录一下对Mogilefs的理解，Mogilefs分为几部分。
0） 数据库（MySQL）部分
你可以用mogdbsetup程序来初始化数据库。数据库保存了Mogilefs的所有元数据，你可以单独拿数据库服务器来做，也可以跟其他程序跑在一起，数据库部分非常重要，类似邮件系统的认证中心那么重要，如果这儿挂了，那么整个Mogilefs将处于不可用状态。因此最好是HA结构。
1）存储节点
mogstored程序的启动将使本机成为一个存储节点。启动时默认去读/etc/mogilefs/mogstored.conf ，具体配置可以参考配置部分。
mogstored启动后，便可以通过mogadm增加这台机器到cluster中。一台机器可以只运行一个mogstored作为存储节点即可，也可以同时运行其他程序。
2）trackers（跟踪器）

mogilefsd即trackers程序，类似mogilefs的wiki上介绍的，trackers做了很多工作，Replication ，Deletion，Query，Reaper，Monitor等等。mogadm,mogtool的所有操作都要跟trackers打交道，Client的一些操作也需要定义好trackers，因此最好同时运行多个trackers来做负载均衡。trackers也可以只运行在一台机器上，也可以跟其他程序运行在一起，只要你配置好他的配置文件即可，默认在/etc/mogilefs/mogilefsd.conf。
3）工具
主要就是mogadm，mogtool这两个工具了，用来在命令行下控制整个mogilefs系统以及查看状态等等。
4）Client
Client实际上是一个Perl的pm，可以写程序调用该pm来使用mogilefs系统，对整个系统进行读写操作。

概念定义
可以参考官方wiki的这儿，简单说一下偶的粗略理解。
domain：最高域，在一个域下key是唯一的。
class：包含在domain中，可以针对每一个class定义保存的份数。
key：对文件的唯一标识。
file：文件。

适用性
由于Mogilefs不支持对一个文件的随机读写，因此注定了只适合做一部分应用。比如图片服务，静态HTML服务。即文件写入后基本上不需要修改的应用，当然你也可以生成一个新的文件覆盖上去。

http://groups.google.com/group/mogile/browse_thread/thread/6a03b1586eb0750c

To install a tracker:
yum -y install mogilefsd perl-MogileFS-Client perl-MogileFS-Utils

To install a storage node:
yum -y install mogstored mogstored-backend-perlbal
mogstored-backend-lighttpd mogstored-backend-apache perl-MogileFS-Client
perl-MogileFS-Utils

Host manipulation:

```
$ mogadm host list
$ mogadm hos
$ mogadm host add foo.local --status=down --ip=10.0.0.34 --port=7900
$ mogadm host mark foo.local down
$ mogadm host modify foo.local --port=7500
$ mogadm host delete foo.local
```

Device manipulation:

```
    $ mogadm device list
    $ mogadm device summary
    $ mogadm device summary --status=dead,down
    $ mogadm device add foo.local 16
    $ mogadm device add foo.local 17 --status=alive
    $ mogadm device mark foo.local 17 down
    $ mogadm device modify foo.local 17 --status=alive --weight=10
    $ mogadm device delete foo.local 17

Domain manipulation:

    $ mogadm domain list
    $ mogadm domain add first.domain
    $ mogadm domain delete first.domain

Class manipulation

    $ mogadm class list
    $ mogadm class add first.domain my.class
    $ mogadm class add first.domain my.class --mindevcount=3
    $ mogadm class modify first.domain my.class --mindevcount=2
    $ mogadm class delete first.domain my.class

Check the status of your entire MogileFS system:

    $ mogadm check

Check MogileFS system statistics:

    $ mogadm stats

Check every file in the entire MogileFS system:

    $ mogadm fsck reset
    $ mogadm fsck start
    $ mogadm fsck status
    $ mogadm fsck printlog

See all the things mogadm can do:

    $ mogadm

Get help on a sub-command:
```

```
    $ mogadm device
```

=head1 CONFIGURATION

It is recommended that you create a configuration file such as C</etc/mogilefs/mogilefs.conf> (or at C<~/.mogilefs.conf>)
to
be used for configuration information.  Basically all you need is something like:

```
    trackers = 10.0.0.23:7001, 10.0.0.15:7001

    # if MogileFS::Admin files aren't installed in standard places:
    lib = /home/mogilefs/cgi-bin
```

Note that these can also be specified on the command line, as per above.

---