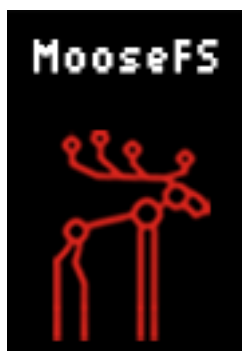


MooseFS

分布式文件系统安装向导



Michał Borychowski
MooseFS Support Manager
contact@moosefs.org

March 2010
Gemius SA

翻译 Translated by: 田逸 (sery@163.com)

北京 Beijing
May 2010

MooseFS 安装向导

翻译：田逸 (sery@163.com)

概述 Overview.....	3
在专用服务器安装 MooseFS 基本步骤	3
主控服务器 Master server 安装	3
备份服务器 Backup server (metallogger) 安装	5
存储块服务器 Chunk servers 安装	5
客户端 Users' computers 安装.....	6
安装 MooseFS 在同一个主机	7
MooseFS 基础用法	10
停止 MooseFS	11

概述 Overview

下面我们以 step-by-step 的方式，介绍在 linux 平台安装 MooseFS 文件系统的基本过程。我们假定您将以 mfs 用户和 mfs 组来运行 MooseFS。同时，我们推荐使用 FHS(文件系统层次结构标准 Filesystem Hierarchy Standard)兼容路径，并且把归档文件 mfs-1.6.15.tar.gz 放置在 /usr/src 目录里。本文旨在向读者介绍怎样在多个专用服务器安装 MooseFS 分布式文件系统以及在单个服务器安装 Moosefs 文件系统用于测试这样的场景。

最新的 MooseFS 稳定发行版本可以从 <http://sourceforge.net/projects/moosefs/> 取得，在安装 MooseFS 系统客户端时，应当确保系统已经安装了正确的 fuse 版本，如果没有 fuse 被安装，您可以从 <http://sourceforge.net/projects/fuse/> 下载并安装它。

在专用服务器安装 MooseFS 基本步骤

我们假定使用的主机 ip 地址分配如下：

- 主控服务器 Master server: 192.168.1.1
- 主控备份服务器 Metalogger server: 192.168.1.2
- 存储块服务器 Chunk servers: 192.168.1.101 and 192.168.1.102
- 客户端主机 (clients): 192.168.2.x

主控服务器 Master server 安装

当我们安装主控服务器时，在配置过程中(./configure)，可以取消安装 chunk server (--disable-mfschunkserver) 以及 MooseFS 客户端 (--disable-mfsmount)。安装主控服务器 master 的具体步骤为：

- 1、添加 mfs 组
#groupadd mfs
- 2、新增系统用户 mfs
#useradd -g mfs mfs
- 3、切换目录
#cd /usr/src
- 4、解包归档文件
#tar -zxvf mfs-1.6.15.tar.gz
- 5、进入安装目录
#cd mfs-1.6.15
- 6、配置
#./configure --prefix=/usr --sysconfdir=/etc \
--localstatedir=/var/lib --with-default-user=mfs \
--with-default-group=mfs --disable-mfschunkserver --disable-mfsmount
- 7、编译并安装
#make
#make install

成功安装 master 以后，系统会在/etc 目录自动生成样例配置文件，这些样例文件是以 .dist 后缀命名。这里我们将借用这些样例文件作为 MooseFS 主控服务器的目标配置文件：

1、切换目录

```
#cd /etc
```

2、复制样例文件，以得到 master 所需的配置文件

```
#cp mfsmaster.cfg.dist mfsmaster.cfg
#cp mfsmetallogger.cfg.dist mfsmetallogger.cfg
#cp mfsexports.cfg.dist mfsexports.cfg
```

如果我们打算更改这些配置文件中的某些项目的值，则需要打开特定配置文件相关文本行前的注释，然后修改跟随其后的值。因为被注释掉的行，即是 **MooseFS** 内置的缺省值。

`mfsmaster.cfg` 配置文件包含主控服务器 **master** 相关的设置，在这里我们暂时不打算对其进行修改。如果你想知道关于该配置文件更多的信息，请查看它的手册页（使用命令 `man mfsmaster.cfg`）。

配置文件 `mfsexports.cfg` 指定那些客户端主机可以远程挂接 **MooseFS** 文件系统，以及授予挂接客户端什么样的访问权限。例如，我们指定只有 **192.168.2.x** 网段的主机可以以读写模式访问 **MooseFS** 的整个共享结构资源（/）。在配置文件 `mfsexports.cfg` 文件的第一行，先取消注释，然后把星号（*）改成 **192.168.2.0/24**，以便我们可以得到下面的文本行：

```
192.168.2.0/24          /      rw,alldirs,maproot=0
```

二进制文件 `metadata` 和文本文件 `changelog` 将被保存在目录 `/var/lib/mfs`，这是因为我们安装过程的 `configure` 步骤使用了选项 `--localstatedir=/var/lib`。首次安装 **master** 时，会自动生成一个名为 `metadata.mfs.empty` 的元数据文件 `metadata`，该文件是空的。**MooseFS master** 运必须有文件 `metadata.mfs`，这个文件是从 `metadata.mfs.empty` 改名而来：

1、切换目录

```
#cd /var/lib/mfs
```

2、重命名文件

```
#cp metadata.mfs.empty metadata.mfs
```

[译者注] **MooseFS master** 运行以后，`metadata.mfs` 文件大小将发生变化，肯定不会是空文件了。

修改 `/etc/hosts` 文件，以绑定主机名 `mfsmaster` 与 ip 地址 **192.168.1.1**：

```
192.168.1.1    mfsmaster
```

这个时候，我们可以试着运行 **master** 服务（服务将以安装配置 `configure` 指定的用户运行，这我们的案例里，它是 `mfs`）：

```
#!/usr/sbin/mfsmaster start
```

在一个生产环境里，我们应当设置自动启动脚本，以便操作系统重新启动时，**MooseFS master** 也能自动运行。

为了监控 **MooseFS** 当前运行状态，我们可以运行 CGI 监控服务，这样就可以用浏览器查看整个 **MooseFS** 的运行情况：

```
#!/usr/sbin/mfscgiserv
```

现在，我们在浏览器地址栏输入 `http://192.168.1.1:9425` 即可查看 **master** 的运行情况（这个时候，是不能看见 `chunk server` 的数据）。

备份服务器 Backup server (metallogger) 安装

用来安装 metallogger 的主机，在性能上应该比 master 强大（至少有更多的内存）。一旦主控服务器 master 失效，只要导入 changelogs 到元数据文件,备份服务器 metallogger 将能接替发生故障的 master，行使管理服务器的职能(更多细节请参看 <http://www.moosefs.org/mini-howtos.html#redundant-master>)。

备份服务器 Metallogger 安装跟主控服务器 master 安装非常类似。其安装命令如下：

- 1、创建组 mfs
#groupadd mfs
- 2、创建用户 mfs
#useradd -g mfs mfs
- 3、切换目录
#cd /usr/src
- 4、解包归档文件
#tar -zxvf mfs-1.6.15.tar.gz
- 5、切换目录
#cd mfs-1.6.15
- 6、配置
#./configure --prefix=/usr --sysconfdir=/etc \
--localstatedir=/var/lib --with-default-user=mfs \
--with-default-group=mfs --disable-mfschunkserver --disable-mfsmount
- 7、编译及安装
#make
#make install
- 8、产生配置文件
#cd /etc
#cp mfsmetallogger.cfg.dist mfsmetallogger.cfg

类似地，修改/etc/hosts 文件，增加下面的行：

```
192.168.1.1    mfsmaster
```

现在，我们来试着运行备份服务 mfsmetallogger:

```
#/usr/sbin/mfsmetallogger start
```

在一个生产环境里，我们应当设置自动启动脚本，以便操作系统重新启动时，MooseFS mfsmetallogger 服务也能自动运行。

存储块服务器 Chunk servers 安装

在每个 chunk server 主机上执行下面的命令：

```
#groupadd mfs  
#useradd -g mfs mfs  
#cd /usr/src  
#tar -zxvf mfs-1.6.15.tar.gz  
#cd mfs-1.6.15
```

```
#./configure --prefix=/usr --sysconffdir=/etc \
--localstatedir=/var/lib --with-default-user=mfs \
--with-default-group=mfs --disable-mfsmaster
#make
#make install
```

准备 **chunk server** 服务所需的配置文件:

```
#cd /etc/
#cp mfschunkserver.cfg.dist mfschunkserver.cfg
#cp mfshdd.cfg.dist mfshdd.cfg
```

为了测试这个安装,我们保留 `mfschunkserver.cfg` 文件不做任何改动;如果读者想了解配置文件 `mfschunkserver.cfg` 更详细的信息,请查看手册页 (`man mfschunkserver.cfg`)。

在配置文件 `mfshdd.cfg` 中,我们给出了用于客户端挂接 **MooseFS** 分布式文件系统根分区所使用的共享空间位置。建议在 **chunk server** 上划分单独的空间给 **MooseFS** 使用,这样做的好处是便于管理剩余空间。此处我们假定要使用两个共享点 `/mnt/mfschunks1` 和

`/mnt/mfschunks2`,为此,我们在 `mfshdd.cfg` 加入下面的文本行:

```
/mnt/mfschunks1
/mnt/mfschunks2
```

在启动 **chunk server** 前,需确保用户 **mfs** 有权限读写将要被挂接的分区(因为 **chunk server** 运行时要在此创建一个 `.lock` 的文件):

```
#chown -R nfs:mfs /mnt/mfschunks1
#chown -R nfs:mfs /mnt/mfschunks2
```

类似地,修改 `/etc/hosts` 文件,增加下面的行:

```
192.168.1.1    mfsmaster
```

开始启动 **chunk server**:

```
#/usr/sbin/mfschunkserver start
```

现在再通过浏览器访问 <http://192.168.1.1:9425/> 应该可以看见这个 **MooseFS** 系统的全部信息,包括主控 **master** 和存储服务 **chunkserver**。

客户端 **Users' computers** 安装

为了挂接基于 **MooseFS** 分布式文件,客户端主机必须安装 **FUSE** 软件包(**fuse** 版本号至少 2.6,推荐使用版本号大于 2.7.2 的 **fuse**)。如果系统没有安装 **fuse**,你必须手动对其进行安装。一种常见的安装方式是从源码进行编译安装-我们可以从 <http://sourceforge.net/projects/fuse/> 取得安装源码:

```
#cd /usr/src
#tar -zxvf fuse-2.8.3.tar.gz
#cd fuse-2.8.3
#./configure
#make
#make install
```

安装客户端软件 `mfsmount` 的步骤:

```
#cd /usr/src
```

```
#tar -zxvf mfs-1.6.15.tar.gz
#cd mfs-1.6.15
#./configure --prefix=/usr --sysconfdir=/etc \
--localstatedir=/var/lib --with-default-user=mfs \
--with-default-group=mfs --disable-mfsmaster \
--disable-mfschunkserver
#make
#make install
```

修改文件/etc/hosts，增加如下的文本行:

```
192.168.1.1    mfsmaster
```

假定客户端的挂接点是/mnt/mfs，我们将以下面的指令来使用 **MooseFS** 分布式共享文件系统:

1、创建挂接点

```
#mkdir -p /mnt/mfs
```

2、开始挂接操作

```
#/usr/bin/mfsmount /mnt/mfs -H mfsmaster
```

执行命令 `df -h | grep mfs` 检查分区情况，可能的输出如下:

```
/storage/mfschunks/mfschunks1
                2.0G   69M   1.9G    4% /mnt/mfschunks1
/storage/mfschunks/mfschunks2
                2.0G   69M   1.9G    4% /mnt/mfschunks2
mfs#mfsmaster:9421   3.2G     0   3.2G    0% /mnt/mfs
```

安装 MooseFS 在同一个主机

如果为测试目的，可以把 **MooseFS** 安装在同一个物理主机上。在这里，我们不推荐您把备份服务也安装在这个主机上。同样，我们假定主机的 ip 地址为 192.168.1.1。

为了挂接基于 **MooseFS** 分布式文件，客户端主机必须安装 FUSE 软件包（fuse 版本号至少 2.6，推荐使用版本号大于 2.7.2 的 fuse）。如果系统没有安装 fuse,你必须手动对其进行安装。一种常见的安装方式是从源码进行编译安装-我们可以从 <http://sourceforge.net/projects/fuse/> 取得安装源码:

```
#cd /usr/src
#tar -zxvf fuse-2.8.3.tar.gz
#cd fuse-2.8.3
#./configure
#make
#make install
```

安装 **MooseFS**:

```
#groupadd mfs
#useradd -g mfs mfs
#cd /usr/src
#tar -zxvf mfs-1.6.15.tar.gz
#cd mfs-1.6.15
#./configure --prefix=/usr --sysconfdir=/etc \
--localstatedir=/var/lib --with-default-user=mfs \
--with-default-group=mfs
```

```
#make
#make install
```

MooseFS chunk 以独占方式使用专门磁盘分区是非常必要的--这样做的好处是便于管理剩余空间。**MooseFS** 并不考虑其剩余空间能被另作他用。如果没有单独创建文件系统的条件，可以在文件中创建一个文件系统。为了完成测试，我们准备两个 **2GB** 的文件（文件位于目录 `/storage/mfschunks`），并在其上创建文件系统。把他们格式化为 **ext3**，分别挂载在 `/mnt/mfschunks1` 和 `/mnt/mfschunks2`。以下是具体操作步骤：

一、挂载第一个文件系统

1、创建目录

```
#mkdir -p /storage/mfschunks
```

2、创建镜像文件 mfschunks1

```
#dd if=/dev/zero of=/storage/mfschunks/mfschunks1 bs=1024 count=1\
seek=$((2*1024*1024-1))
```

3、创建文件系统

```
#mkfs -t ext3 /storage/mfschunks/mfschunks1
```

4、创建挂载点

```
#mkdir -p /mnt/mfschunks1
```

5、挂载文件系统

```
#mount -t ext3 -o loop /storage/mfschunks/mfschunks1\
/mnt/mfschunks1
```

二、挂载第二个文件系统

1、创建第二个镜像文件

```
#dd if=/dev/zero of=/storage/mfschunks/mfschunks2 bs=1024 count=1\
seek=$((2*1024*1024-1))
```

2、创建文件系统

```
#mkfs -t ext3 /storage/mfschunks/mfschunks2
```

3、创建挂载点

```
#mkdir -p /mnt/mfschunks2
```

4、挂载文件系统

```
#mount -t ext3 -o loop /storage/mfschunks/mfschunks2 \
/mnt/mfschunks2
```

在启动 **chunk server** 前，需确保用户 **mfs** 有权限读写将要被挂载的分区（因为 **chunk server** 运行时要在此创建一个 **.lock** 的文件）：

```
#chown -R nfs:mfs /mnt/mfschunks1
#chown -R nfs:mfs /mnt/mfschunks2
```

`/etc` 目录自动生成样例配置文件，这些样例文件是以 **.dist** 后缀命名。这里我们将借用这些样例文件作为 **MooseFS** 的目标配置文件：

```
#cd /etc
#cp mfsexports.cfg.dist mfsexports.cfg
#cp mfsmaster.cfg.dist mfsmaster.cfg
#cp mfschunkserver.cfg.dist mfschunkserver.cfg
#cp mfshdd.cfg.dist mfshdd.cfg
```

mfsexports.cfg 与 **mfsmaster.cfg** 为主控服务 **master** 配置文件，**mfschunkserver.cfg** 与 **mfshdd.cfg** 为 **chunk server** 配置文件。

配置文件 `mfsexports.cfg` 指定那些客户端主机可以远程挂载 **MooseFS** 文件系统，以及授予挂载客户端什么样的访问权限。例如，我们指定只有 **192.168.2.x** 网段的主机可以以读写模式访问 **MooseFS** 的整个共享结构资源 (/)。在配置文件 `mfsexports.cfg` 文件的第一行，先取消注释，然后把星号 (*) 改成 **192.168.1.0/24**，以便我们可以得到下面的文本行：

```
192.168.1.0/24          /      rw,alldirs,maproot=0
```

修改配置文件 `mfshdd.cfg`，使其内容为：

```
/mnt/mfschunks1
/mnt/mfschunks2
```

作为测试的例子，我们不打算修改 `mfsmaster.cfg` 和 `mfschunkserver.cfg` 配置文件的其他选项。

二进制文件 `metadata` 和文本文件 `changelog` 将被保存在目录 `/var/lib/mfs`，这是因为我们安装过程的 `configure` 步骤使用了选项 `--localstatedir=/var/lib`。首次安装 `master` 时，会自动生成一个名为 `metadata.mfs.empty` 的元数据文件 `metadata`，该文件是空的。**MooseFS** `master` 运必须有文件 `metadata.mfs`，这个文件是从 `metadata.mfs.empty` 改名而来：

```
#cd /var/lib/mfs
#cp metadata.mfs.empty metadata.mfs
```

修改文件 `/etc/hosts`，新增如下的文本行：

```
192.168.1.1    mfsmaster
```

运行 `master server`，`CGI` 监控以及 `chunk server`：

```
#!/usr/sbin/mfsmaster start
#!/usr/sbin/mfscgiserv
#!/usr/sbin/mfschunkserver start
```

MooseFS 当前运行状态可以在浏览器中地址栏输入 <http://192.168.1.1:9425/> 获得。

挂载 **MooseFS** 文件系统到挂载点 `/mnt/mfs`：

1、建立挂载点

```
#mkdir -p /mnt/mfs
```

2、挂载操作

```
#!/usr/bin/mfsmount /mnt/mfs -H mfsmaster
```

3、查看挂载情况

```
#df -h | grep mfs:
/storage/mfschunks/mfschunks1
                2.0G   69M   1.9G    4% /mnt/mfschunks1
/storage/mfschunks/mfschunks2
                2.0G   69M   1.9G    4% /mnt/mfschunks2
mfs#mfsmaster:9421  3.2G     0   3.2G    0% /mnt/mfs
```

MooseFS 基础用法

在 MooseFS 挂接点下创建目录 **folder1**, 在该目录, 我们将以一个副本的方式存放文件 (设置 `goal=1`):

```
#mkdir -p /mnt/mfs/folder1
```

再在挂接点创建第 2 个目录 **folder2**, 在该目录, 我们将以两个副本的方式存放文件(设置 `goal=2`):

```
#mkdir -p /mnt/mfs/folder2
```

使用命令 `mfssetgoal -r` 设定目录里文件的副本数:

1、副本数为 1

```
#mfssetgoal -r 1 /mnt/mfs/folder1
```

```
/mnt/mfs/folder1:
```

inodes with goal changed:	0
inodes with goal not changed:	1
inodes with permission denied:	0

2、副本数为 2

```
#mfssetgoal -r 2 /mnt/mfs/folder2
```

```
/mnt/mfs/folder2:
```

inodes with goal changed:	0
inodes with goal not changed:	1
inodes with permission denied:	0

拷贝同一个文件到两个目录:

```
cp /usr/src/mfs-1.6.15.tar.gz /mnt/mfs/folder1
```

```
cp /usr/src/mfs-1.6.15.tar.gz /mnt/mfs/folder2
```

命令 `mfschunkfile` 用来检查给定的文件以多少副本数来存储。对应目录 **folder1** 来说, 有一个副本存储在一个 chunk 里:

```
#mfscheckfile /mnt/mfs/folder1/mfs-1.6.15.tar.gz
```

```
/mnt/mfs/folder1/mfs-1.6.15.tar.gz:
```

```
1 copies: 1 chunks
```

而在目录 **folder2** 中, 文件 `mfs-1.6.15.tar.gz` 是以两个副本保存的:

```
#mfscheckfile /mnt/mfs/folder2/mfs-1.6.15.tar.gz
```

```
/mnt/mfs/folder2/mfs-1.6.15.tar.gz:
```

```
2 copies: 1 chunks
```

附加信息。当所有的组件被安装到同一个物理主机的时候, 即便设定了 `goal=2` 来达到保存两个副本的目的, 但你可能看到的只是一个副本而已—这是合理的, 尽管有两个磁盘, 但它只是一个 **chunk server** 啊!

更多关于 MooseFS 命令的使用方法, 可以在这里找到:

<http://www.moosefs.org/reference-guide.html#using-moosefs>

我们推荐你阅读 FAQ 页面内容:

<http://www.moosefs.org/moosefs-faq.html>

停止 MooseFS

为了安全停止 MooseFS 集群，建议执行如下的步骤:

- 在所有客户端用 `Unmount` 命令先卸载文件系统(本例将是: `umount /mnt/mfs`)
- 停止 `chunk server` 进程: `/usr/sbin/mfschunkserver stop`
- 停止 `metallogger` 进程: `/usr/sbin/mfsmetallogger stop`
- 停止主控 `master server` 进程: `/usr/sbin/mfsmaster stop`

北京 Beijing
2010/5/5