

目 录

1	分布式文件系统 Mogilefs 介绍	1
2	安装 Mogilefs	2
2.1	服务器配置	2
2.2	安装 Mysql 和 Track	2
2.3	安装 Storage	3
2.4	安装 Nginx	4
3	WEB API	6
3.1	上传文件	6
3.2	删除文件	7
3.3	下载文件	7

更新日期	作者	URL
2012-8-27	风河	www.nsbeta.info

1 分布式文件系统 Mogilefs 介绍

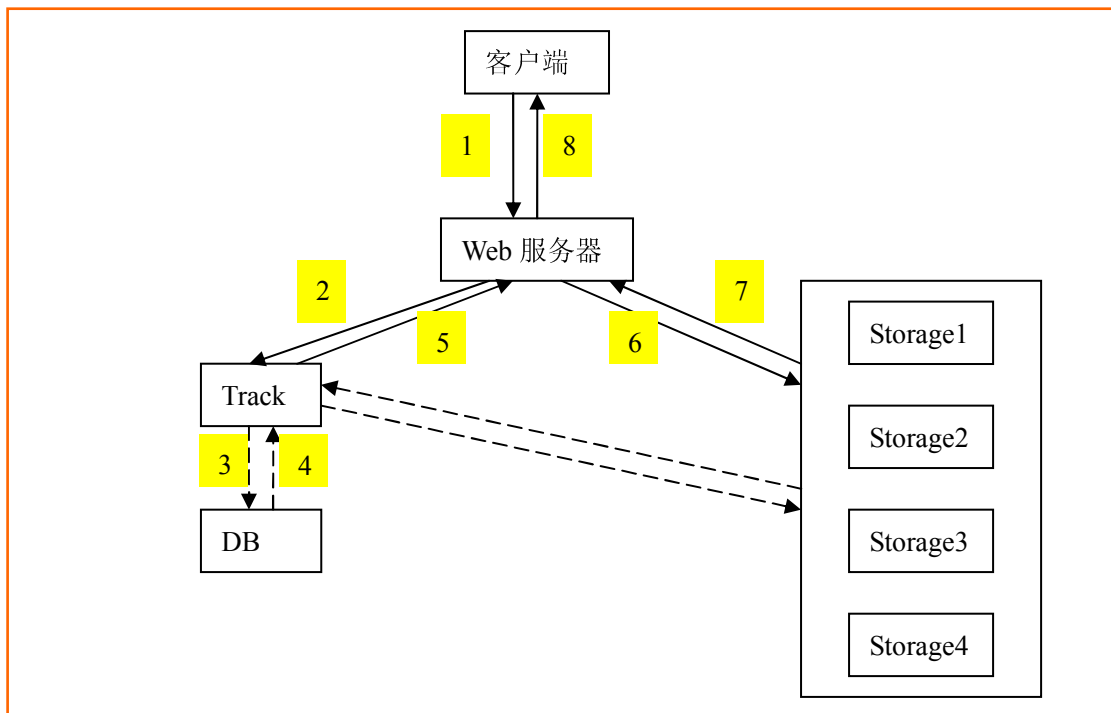
Mogilefs 是很早的分布式文件系统，我 05 年时用过一次，当时给 T-Mobile 做了一个小项目，用到了这个东西。在本文写作时，看了下 [MogileFS-Server-2.65](#) 的最近发布日期是 2012 年 8 月 14 日，说明该项目还是被积极维护的。Mogilefs 最初是由 Danga 写的，Danga 的另一产品大名鼎鼎，就是 Memcached。

Mogilefs 的结构非常简单，分为四个部分：

- Track
- 数据库
- Storage
- WEB API

Track 是管理中心和调度器，它存储服务配置信息和文件索引，例如某个文件存储在哪个节点的哪个设备上。数据库是 Track 的后台 DB，默认支持 Mysql，表结构非常简单，每个表有什么作用一看就懂。Storage 是存储节点，接受客户端上传或下载文件。WEB API 是客户可以通过 Web 服务器以 HTTP 的方式来访问 Mogilefs，包括上传、下载、删除文件。

客户端访问 Mogilefs 的流程图大概如下：



第 1 步，客户端对 Web 服务器发起请求，例如上传或下载文件请求

第 2 步，Web 服务器询问 Track 关于系统的配置信息，例如可用存储节点、或已有文件的存储位置

第 3 步，Track 查询数据库，询问系统配置信息

第 4 步，数据库返回结果，例如文件存储在哪个节点

第 5 步, Track 将结果告知 Web 服务器

第 6 步, Web 服务器请求特定的 Storage 节点, 这个其实也是 HTTP 请求

第 7 步, Storage 返回结果给 Web 服务器

第 8 步, Web 服务器将结果转发给客户端

2 安装 Mogilefs

2.1 服务器配置

服务名	IP 地址	进程名	配置文件	服务端口
Track	192.168.0.114	mogilefsd	/etc/mogilefs/mogilefsd.conf	7001
Mysql	192.168.0.114	mysqld	/etc/mysql/my.cnf	3306
Nginx	192.168.0.114	nginx	/usr/local/nginx/conf/nginx.conf	80
Storage	192.168.0.176 192.168.0.177	mogstored	/etc/mogilefs/mogstored.conf	7500 7501

服务器的操作系统都是 Ubuntu Linux Server

2.2 安装 Mysql 和 Track

1. apt-get install mysql-server

2. 装好数据库后, 执行 SQL:

```
create database mogilefs;
```

```
grant all privileges on mogilefs.* to 'mogile'@'%' identified by '***';
```

建好数据库及用户密码。

3. 运行 perl -MCPAN -e shell, 在出来的 shell 里运行:

```
install App::cpanminus
```

安装 cpanm

4. 运行如下 2 个命令, 安装 mogilefs 程序:

```
cpanm MogileFS::Server
```

```
cpanm MogileFS::Utils
```

5. 运行如下命令, 创建 mogilefs 的数据库表结构:

```
mogdbsetup --dbname=mogilefs --dbuser=mogile --dbpassword=***
```

将***替换为正确密码。

6. 创建/etc/mogilefs/mogilefsd.conf 配置文件, 内容如下:

```
db_dsn = DBI:mysql:mogilefs:host=localhost;port=3306;mysql_connect_timeout=5
```

```
db_user = mogile
```

```
db_pass = ***
conf_port = 7001
listener_jobs = 5
node_timeout = 5
rebalance_ignore_missing = 1
```

7. 因为 Track 不能以 root 运行，所以创建一个 mogile 用户

```
mkdir -p /home/mogile
useradd mogile -d /home/mogile -s /bin/bash
```

8. 启动 Track 程序：

```
# su - mogile
$ mogilefsd -c /etc/mogilefs/mogilefsd.conf --daemon
$ exit
```

2.3 安装 Storage

1. 跟安装 Track 一样，在存储节点上同样安装 mogilefs 程序：

```
cpanm MogileFS::Server
cpanm MogileFS::Utils
```

2. 创建分布式文件系统的存储目录：

```
mkdir /data/mogdata
```

3. 创建磁盘设备：

在一台存储节点上，执行：

```
mkdir /data/mogdata/dev{1,2}
```

另一台存储节点上，执行：

```
mkdir /data/mogdata/dev{3,4}
```

至少存在 2 个存储节点服务器，每个服务器至少创建 2 个磁盘设备。

正常情况下，mogilefs 会将每份文件，分别写往 2 个存储节点服务器的其中一个设备。

- 如果一个存储节点服务器挂了，那么 mogilefs 会将文件写往另一台存储节点服务器上的 2 个磁盘设备。

4. 注册存储节点服务器之一：

```
mogadm --trackers=192.168.0.114:7001 host add mogstore1 --ip=192.168.0.176 --port=7500
--status=alive
```

注册磁盘设备：

```
mogadm --trackers=192.168.0.114:7001 device add mogstore1 1
```

```
mogadm --trackers=192.168.0.114:7001 device add mogstore1 2
```

上述 mogstore1 是主机名，必须全局唯一。

device 里的 1 和 2 是设备名，必须全局唯一。

5. 注册存储节点服务器之二：

```
mogadm --trackers=192.168.0.114:7001 host add mogstore2 --ip=192.168.0.177 --port=7500
--status=alive
```

注册磁盘设备:

```
mogadm --trackers=192.168.0.114:7001 device add mogstore2 3
```

```
mogadm --trackers=192.168.0.114:7001 device add mogstore2 4
```

上述 mogstore2 是主机名, 必须全局唯一。

device 里的 3 和 4 是设备名, 必须全局唯一。

6. 创建配置文件/etc/mogilefs/mogstored.conf, 如下内容:

```
httplisten=0.0.0.0:7500
```

```
mgmtlisten=0.0.0.0:7501
```

```
docroot=/data/mogdata
```

7. 以 root 启动存储服务:

```
# mogstored --daemon
```

8. 检查系统是否正常, 命令及输出如下:

```
# mogadm --trackers=192.168.0.114:7001 check
```

```
Checking trackers...
```

```
192.168.0.114:7001 ... OK
```

```
Checking hosts...
```

```
[ 1] mogstore1 ... OK
```

```
[ 2] mogstore2 ... OK
```

```
Checking devices...
```

host device	size(G)	used(G)	free(G)	use%	ob state	I/O%
[1] dev1	1707.511	4.659	1702.853	0.27%	writable	0.0
[1] dev2	1707.511	4.659	1702.853	0.27%	writable	0.0
[2] dev3	1707.511	174.004	1533.507	10.19%	writable	0.0
[2] dev4	1707.511	174.004	1533.507	10.19%	writable	0.0
total:	6830.045	357.326	6472.719	5.23%		

2.4 安装 Nginx

1. 需要三个文件:

- (1) nginx-1.2.2.tar.gz 当然更高版本亦可

- (2) nginx_mogilefs_module-1.0.4.tar.gz 针对 mogilefs 的 nginx 模块

- (3) mogilefs_count.patch 上述模块有个 bug, 这个是补丁

2. 解开上述文件。

3. cd nginx_mogilefs_module-1.0.4/
patch < ../mogilefs_count.patch
打好补丁
4. cd nginx-1.2.2
./configure --prefix=/usr/local/nginx-1.2.2 --add-module=../nginx_mogilefs_module-1.0.4
make && make install
安装 nginx 和 mogilefs 模块
5. 在 mogilefs 里增加一个域:
mogadm --trackers=192.168.0.114:7001 domain add testdfs

6. 创建 nginx 的站点配置文件如下:

```
server {  
    listen      80;  
    server_name dfs.test.com;  
  
    charset utf-8;  
    ssi on;  
    access_log /data/log/nginx/dfs.access.log;  
  
    location /download/ {  
        mogilefs_tracker 192.168.0.114:7001;  
        mogilefs_domain testdfs;  
  
        mogilefs_pass {  
            proxy_pass $mogilefs_path;  
            proxy_hide_header Content-Type;  
            proxy_buffering off;  
        }  
    }  
  
    location /upload/ {  
        allow 192.168.0.0/24;  
        deny all;  
  
        mogilefs_tracker 192.168.0.114:7001;  
        mogilefs_domain testdfs;  
        mogilefs_methods PUT DELETE;  
  
        mogilefs_pass {  
            proxy_pass $mogilefs_path;  
            proxy_hide_header Content-Type;  
            proxy_buffering off;  
        }  
    }  
}
```

```

    }
}

error_page 500 502 503 504 /50x.html;
location = /50x.html {
    root html;
}
}

```

上述配置文件里，主要是定义了/download/和/upload/的配置，顾名思义就是下载和上传的配置，如果 URL 路径含有这 2 个关键字，就由 mogilefs 模块处理。

/upload/那里的 allow 配置涉及到安全，必须控制好。

mogilefs_domain 是预先定义的分布式文件系统域，这里是 testdfs。

7. 启动 nginx

3 WEB API

3.1 上传文件

curl --upload-file meinv.jpg <http://dfs.test.com/upload/meinv.jpg>

注意：/upload/路径后的文件名称（meinv.jpg），就是 mogilefs 的索引 key 名称。

上传多个不同的文件，这个 key 必须保证全局唯一，否则新的就会覆盖旧的。

curl 的—upload-file 参数，实际是发送 PUT 这个 HTTP 头部方法，相当于 curl -X PUT 如果编程调用，就要在程序里指定请求 HTTP 方法是 PUT，而不是 GET 或 POST。

上传后，如果了解文件存储在哪个设备上，登陆数据库查询。例如：

```

mysql> select dkey as file,devid from file_on left join file
-> on file_on.fid=file.fid where dkey='meinv.jpg';
+-----+-----+
| file      | devid |
+-----+-----+
| meinv.jpg | 1     |
| meinv.jpg | 4     |
+-----+-----+
2 rows in set (0.00 sec)

```

上述说明这个文件位于设备 1 和设备 4 上。

3.2 删除文件

`curl -X DELETE http://dfs.test.com/upload/meinv.jpg`

如果是编程调用，在程序里指定请求 HTTP 方法是 DELETE。

3.3 下载文件

直接 GET 访问：<http://dfs.test.com/download/meinv.jpg>

注意路径是 download 而不是上传和删除的 upload