



1 FastDFS 安装

1.1 安装 VMware 虚拟机

安装 CentOS7 虚拟机。

1.2 FastDFS--tracker 安装

分别在 192.168.101.3 和 192.168.101.4 上安装 tracker。

注：初次安装可只安装一台 tracker，快速体验。

1.2.1 下载

tracker 和 storage 使用相同的安装包，下载地址：

<https://github.com/happyfish100/FastDFS>

本教程下载：`FastDFS_v5.05.tar.gz`

1.2.2 FastDFS 安装环境

FastDFS 是 C 语言开发，建议在 linux 上运行，本教程使用 Centos7 作为安装环境。

安装 FastDFS 需要先将官网下载的源码进行编译，编译依赖 gcc 环境，如果没有 gcc 环境，需要安装 `gcc : yum install gcc-c++`



1.2.3 安装 libevent

FastDFS 依赖 libevent 库，需要安装：

```
yum -y install libevent
```

1.2.4 安装 libfastcommon

libfastcommon 是 FastDFS 官方提供的，libfastcommon 包含了 FastDFS 运行所需要的一些基础库。

将 libfastcommonV1.0.7.tar.gz 拷贝至/usr/local/下

```
cd /usr/local
```

```
tar -zxvf libfastcommonV1.0.7.tar.gz
```

```
cd libfastcommon-1.0.7
```

```
./make.sh
```

```
./make.sh install
```

注意：libfastcommon 安装好后会自动将库文件拷贝至/usr/lib64 下，由于 FastDFS 程序引用 usr/lib 目录所以需要将/usr/lib64 下的库文件拷贝至/usr/lib 下。

要拷贝的文件如下：

或者建立软连接：

```
ln -s /usr/lib64/libfastcommon.so /usr/lib/libfastcommon.so
```

```
[root@server01 libfastcommon-1.0.7]# ll /usr/lib64
总用量 256
-rwxr-xr-x. 1 root root 260053 5月 18 00:56 libfastcommon.so
```

第二个软连接(local 这个路径好像不需要，费解)

```
ln -s /usr/lib64/libfastcommon.so /usr/local/lib/libfastcommon.so
```



1.2.5 tracker 编译安装

将 FastDFS_v5.05.tar.gz 拷贝至/usr/local/下

```
tar -zxvf FastDFS_v5.05.tar.gz
```

```
cd FastDFS
```

```
./make.sh 编译
```

```
./make.sh install 安装
```

安装成功将安装目录下的 conf 下的文件拷贝到/etc/fdfs/下。

```
client.conf
http.conf
mime.types
storage.conf
storage_ids.conf
tracker.conf
```

1.2.6 配置 tracker 配置文件

安装成功后进入/etc/fdfs 目录：

```
[root@server01 FastDFS]# cd /etc/fdfs/
[root@server01 fdfs]# ll
总用量 20
-rw-r--r--. 1 root root 1461 5月 18 00:58
-rw-r--r--. 1 root root 7829 5月 18 00:58
-rw-r--r--. 1 root root 7102 5月 18 00:58
```

```
# the method of selecting group to upload files
# 0: round robin
# 1: specify group
# 2: load balance, select the max free space group to upload file
store_lookup=2

# which group to upload file
# when store_lookup set to 1, must set store_group to the group name
store_group=group1

# which storage server to upload file
# 0: round robin (default)
# 1: the first server order by ip address
# 2: the first server order by priority (the minimal)
store_server=1

# which path(means disk or mount point) of the storage server to upload file
# 0: round robin
# 2: load balance, select the max free space path to upload file
store_path=2

# which storage server to download file
# 0: round robin (default)
# 1: the source storage server which the current file uploaded to
download_server=1
```

这几个参数也对下载有影响

拷贝一份新的 tracker 配置文件：

```
cp tracker.conf.sample tracker.conf
```

修改 tracker.conf

```
cd /etc/fdfs/
删除当前目录下所有文件
拷贝FastDFS安装目录下conf下面所有文件到当前目录
cp /usr/local/fastDFS/FastDFS/conf/* ./
编辑配置文件目录下的tracker.conf, 一般只需改动以下几个参数即可:
base_path=/fdfs/tracker      #设置tracker的数据文件和日志目录
max_connections=300           #最大连接数
work_threads=1                 #工作线程数, 最好和cpu核数相同
http.server_port=8080          #设置http端口号
```

```
vi tracker.conf
```

base_path=/usr/local/FastDFS
是\$home\$目录, 看自己安装在哪一个目录



查看fdisk的运行情况

```
/usr/bin/fdfs_monitor /etc/fdfs/client.conf
```

5. 开机启动

```
echo '/usr/bin/fdfs_storaged /etc/fdfs/storage.conf restart' >> /etc/rc.d/rc.local
```

6. 是否启动

```
netstat -tupln | grep storaged
```

1.2.7 启动与检查

这个命令在fdisk里面也能找到

```
/usr/bin/fdfs_trackerd /etc/fdfs/tracker.conf restart 这里是两个文件
```

启动的日志显示先停止 5619 进程（实际环境不是 5619）再启动，如下图：

```
[root@server01 conf]# /usr/bin/fdfs_trackerd /etc/fdfs/tracker.conf restart
waiting for pid [5619] exit ...
starting ...
```

注意：如果没有显示上图要注意是否正常停止原有进程。

检查启动进程：

```
ps -ef | grep fdfs          root      60894      1  0 01:04 ?          00:00:00 /usr/bin/fdfs_trackerd /etc/fdfs/tracker.conf
```

```
                           root      60971      1  0 01:07 ?          00:00:00 /usr/bin/fdfs_storaged /etc/fdfs/storage.conf
```

检查启动端口：

```
netstat -aopn | grep fdfs          root      60985  43839  0 01:08 pts/0          00:00:00 grep fdfs
```

```
tcp      0      0 0.0.0.0:23000          0.0.0.0:*          LISTEN      60971/fdfs_storaged off (0.00/0/0)
tcp      0      0 0.0.0.0:22122          0.0.0.0:*          LISTEN      60894/fdfs_trackerd off (0.00/0/0)
tcp      0      0 192.168.18.178:22122      192.168.18.178:42090      ESTABLISHED 60894/fdfs_trackerd keepalive (61.
tcp      0      0 192.168.18.178:42090      192.168.18.178:22122      ESTABLISHED 60971/fdfs_storaged keepalive (61.
```

1.3 FastDFS--storage 安装

分别在 192.168.101.5、192.168.101.6、192.168.101.7、192.168.101.8 上安装 storage。

注：初次安装可只安装一台 storage，快速体验。



1.3.1 安装 libevent

同 tracker 安装

1.3.2 安装 libfastcommon

同 tracker 安装。

1.3.3 storage 编译安装

同 tracker 编译安装。

1.3.4 配置storage 配置文件

安装成功后进入/etc/fdfs 目录：

```
[root@server01 FastDFS]# cd /etc/fdfs/
[root@server01 fdfs]# ll
总用量 20
-rw-r--r--. 1 root root 1461 5月 18 00:58 client.conf.sample
-rw-r--r--. 1 root root 7829 5月 18 00:58 storage.conf.sample
-rw-r--r--. 1 root root 7102 5月 18 00:58 tracker.conf.sample
```

拷贝一份新的 storage 配置文件：

```
cp storage.conf.sample storage.conf
```

修改 storage.conf

vi storage.conf

```
base_path=/usr/local/FastDFS
store_path0=/usr/local/FastDFS/fdfs_storage0
tracker_server=192.168.72.129:22122
```

```
cd /etc/fdfs/
删除当前目录下所有文件
拷贝FastDFS安装目录下conf下面所有文件到当前目录
cp /usr/local/fastDFS/FastDFS/conf/* ./
编辑配置文件目录下的storage.conf，一般只需改动以下几个参数即可。
disabled=false          #启用配置文件
max_connections=300      #最大连接数
work_threads=1           #工作线程数，最好和cpu核数相同
group_name=head          #组名，目前我们有3个(head、item、other)
port=23000               #设置storage的端口号(head、item、other)
base_path=/fdfs/storage  #设置storage的日志目录（需预先创建）
store_path_count=1        #存储路径个数，需要和store_path个数匹配
store_path0=/fdfs/head    #存储路径
tracker_server=192.168.31.128:22122 #tracker服务器的IP地址和端口号
http.server_port=8080      #设置http端口号
```

如果有多个挂载磁盘则定义多个store_path，如下

```
#store_path1=....
```

```
#store_path2=....
```

如果有多个tracker则配置多个tracker_server



1.3.5 启动

```
/usr/bin/fdfs_storaged /etc/fdfs/storage.conf restart
```

启动的日志显示先停止 8931 进程（实际环境不是 8931）再启动，如下图：

```
[root@server01 data]# /usr/bin/fdfs_storaged /etc/fdfs/storage.conf restart
waiting for pid [8931] exit ...
starting ...
```

注意：如果没有显示上图要注意是否正常停止原有进程。



1.4 上传图片测试

1.4.1 通过 fdfs_test 程序

FastDFS 安装成功可通过/usr/bin/fdfs_test 程序来测试上传、下载等操作。

修改/etc/fdfs/client.conf

tracker_server 根据自己部署虚拟机的情况配置。

```
base_path=/home/fastdfs
tracker_server=192.168.101.3:22122
tracker_server=192.168.101.4:22122
```

使用格式：

/usr/bin/fdfs_test 客户端配置文件地址 upload 上传文件

比如将/home 下的图片上传到 FastDFS 中：

/usr/bin/fdfs_test /etc/fdfs/client.conf upload /home/tomcat.png

打印如下日志：

```
This is FastDFS client test program v5.05
```

```
Copyright (C) 2008, Happy Fish / YuQing
```



FastDFS may be copied only under the terms of the GNU General Public License V3, which may be found in the FastDFS source kit. Please visit the FastDFS Home Page <http://www.csource.org/> for more detail.

[2015-05-18 02:07:10] DEBUG - base_path=/home/FastDFS, connect_timeout=30, network_timeout=60, tracker_server_count=1, anti_steal_token=0, anti_steal_secret_key length=0, use_connection_pool=0, g_connection_pool_max_idle_time=3600s, use_storage_id=0, storage server id count: 0

tracker_query_storage_store_list_without_group:

server 1. group_name=, ip_addr=192.168.101.5, port=23000

server 2. group_name=, ip_addr=192.168.101.6, port=23000

group_name=group1, ip_addr=192.168.101.5, port=23000

storage_upload_by_filename

group_name=group1,

remote_filename=M00/00/00/wKhIBVVY2M-AM_9DAAAT7-0xdqM485.png

source ip address: 192.168.101.5

file timestamp=2015-05-18 02:07:11

file size=5103



```
file crc32=3979441827
```

```
example
```

```
file
```

```
url:
```

```
http://192.168.101.5/group1/M00/00/00/wKhIBVVY2M-AM_9DAAAT7-0xdqM485.
```

```
png
```

```
storage_upload_slave_by_filename
```

```
group_name=group1,
```

```
remote_filename=M00/00/00/wKhIBVVY2M-AM_9DAAAT7-0xdqM485_big.png
```

```
source ip address: 192.168.101.5
```

```
file timestamp=2015-05-18 02:07:11
```

```
file size=5103
```

```
file crc32=3979441827
```

```
example
```

```
file
```

```
url:
```

```
http://192.168.101.5/group1/M00/00/00/wKhIBVVY2M-AM_9DAAAT7-0xdqM485_
```

```
big.png
```

http://192.168.101.5/group1/M00/00/00/wKhIBVVY2M-AM_9DAAAT7-0xdqM485_

big.png 就是文件的下载路径。

对应 storage 服务器上的

/home/fastdfs/fdfs_storage/data/00/00/wKhIBVVY2M-AM_9DAAAT7-0xdqM485_b
ig.png 文件。



由于现在还没有和 nqinx 整合无法使用 http 下载。

example file url是不能在浏览器中直接打开的，除非配合nginx使用。

使用`fdns_delete_file`来删除文件，使用格式如下：

`fdns_delete_file /etc/fdfs/client.conf group1/M00/00/00/wKgSslVklFOAWOw_AADfP3RIu5Y687_big.jpg`

删除文件需要完整的group_name和remote_filename。

1.5 FastDFS 和 nginx 整合

1.5.1 主机设置nginx 代理,虚拟主机

单独安装 nginx 代理服务，它的作用是代理访问 storage 上的文件，实现负载均衡。

nginx 的安装细节参考 nginx 文档，这里使用单机 nginx，也可以使用两台 nginx 组成高可用或者采用lvs+nginx访问 Storage 上的 nginx。

创建 nginx-fdfs.conf 配置文件：

```
#storage 群 group1 组

upstream storage_server_group1{
    server 192.168.101.5:80 weight=10;
    server 192.168.101.6:80 weight=10;
}

#storage 群 group2 组

upstream storage_server_group2{
    server 192.168.101.7:80 weight=10;
    server 192.168.101.8:80 weight=10;
}
```



```
server {  
  
    listen 80;  
  
    server_name ccc.test.com;  
  
  
    location /group1{  
  
        proxy_redirect off; 这是什么意思  
  
        proxy_set_header Host $host;  
  
        proxy_set_header X-Real-IP $remote_addr;  
  
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
  
        proxy_pass http://storage_server_group1;  
  
    }  
  
    location /group2{  
  
        proxy_redirect off;  
  
        proxy_set_header Host $host;  
  
        proxy_set_header X-Real-IP $remote_addr;  
  
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;  
  
        proxy_pass http://storage_server_group2;  
  
    }  
  
}
```



1.5.2 在 Storage 上安装 nginx

在 storage server 上安装 nginx 的目的是对外通过 http 访问 storage server 上的文件。 使用 nginx 的模块 FastDFS-nginx-module 的作用是通过 http 方式访问 storage 中的文件，当 storage 本机没有要找的文件时向源 storage 主机代理请求文件。

下载好nginx：<http://nginx.org/en/download.html>
下载好fastdfs-nginx-module：<https://github.com/happyfish100/fastdfs-nginx-module>

1.5.2.1 FastDFS-nginx-module

将 FastDFS-nginx-module_v1.16.tar.gz 传至 fastDFS 的 storage 服务器的 /usr/local/ 下，执行如下命令：

```
cd /usr/local
```

```
tar -zxvf FastDFS-nginx-module_v1.16.tar.gz
```

```
cd FastDFS-nginx-module/src
```

修改 config 文件将 /usr/local/ 路径改为 /usr/

```
ngx addon name=ngx http fastdfs module
HTTP_MODULES="$HTTP_MODULES ngx_http_fastdfs_module"
NGX_ADDON_SRCS="$NGX_ADDON_SRCS $ngx_addon_dir/nginx_http_fastdfs_module.c"
CORE_INCS="$CORE_INCS /usr/include/fastdfs /usr/include/fastcommon/"
CORE_LIBS="$CORE_LIBS -L/usr/lib -lfastcommon -lfdfsclient"
CFLAGS="$CFLAGS -D_FILE_OFFSET_BITS=64 -DFDFS_OUTPUT_CHUNK_SIZE='256*1024' -DFDFS_MOD_CONF_FILENAME='\"/etc/fdfs/mod_fastdfs.conf\"'"
```

将 FastDFS-nginx-module/src 下的 mod_FastDFS.conf 拷贝至 /etc/fdfs/ 下

```
cp mod_FastDFS.conf /etc/fdfs/
```

并修改 mod_FastDFS.conf 的内容：

```
vi /etc/fdfs/mod_FastDFS.conf
```

```
base_path=/home/FastDFS 日志
```



```
tracker_server=192.168.101.3:22122
tracker_server=192.168.101.4:22122
url_have_group_name=true      #url 中包含 group 名称
store_path0=/home/fastdfs/fdfs_storage    #指定文件存储路径
                               /usr/local/FastDFS/fdfs_storage0
#如果有多个
```

将 libfdfsclient.so 拷贝至/usr/lib 下

```
cp /usr/lib64/libfdfsclient.so /usr/lib/
```

创建 nginx/client 目录

```
mkdir -p /var/temp/nginx/client
```

1.5.2.2 nginx 安装

将 nginx-1.8.0.tar.gz 拷贝到/usr/local 下

解压 nginx-1.8.0.tar.gz

进入 nginx-1.8.0 目录，执行如下配置命令：

下边红色部分表示添加 FastDFS-nginx-module 模块

```
./configure \
--prefix=/usr/local/nginx \
--pid-path=/var/run/nginx/nginx.pid \
```



```
--lock-path=/var/lock/nginx.lock \
--error-log-path=/var/log/nginx/error.log \
--http-log-path=/var/log/nginx/access.log \
--with-http_gzip_static_module \
--http-client-body-temp-path=/var/temp/nginx/client \
--http-proxy-temp-path=/var/temp/nginx/proxy \
--http-fastcgi-temp-path=/var/temp/nginx/fastcgi \
--http uwsgi-temp-path=/var/temp/nginx/uwsgi \
--http-scgi-temp-path=/var/temp/nginx/scgi \
--add-module=/usr/local/FastDFS-nginx-module/src
```

[添加模块](#)

make

[这里是覆盖安装](#)

make install

1.5.2.3 nginx 配置文件

新建一个 nginx 配置文件 nginx-fdfs.conf.

[用另一个配置文件配置](#)

添加虚拟主机:

```
server {
    listen      80;
```



```
server_name 192.168.101.65;

location /group1/M00/{
    用root是不能找到文件的, alias好一点, 不然每次都请求tracker
    root /home/FastDFS/fdfs_storage/data;

    ngx_FastDFS_module;

}

location /group{
    # alias /usr/local/FastDFS/fdfs_storage1/data/;
    #没有的话由这个模块tracker请求
    ngx_fastdfs_module;
}
```

需要加这个

说明：

server_name 指定本机 ip

location /group1/M00/ : group1 为 nginx 服务 FastDFS 的分组名称 , M00 是 FastDFS

自动生成编号 , 对应 store_path0=/home/FastDFS/fdfs_storage , 如果 FastDFS 定义
store_path1 , 这里就是 M01

启动nginx : /usr/local/nginx/sbin/nginx -c /usr/local/nginx/conf/nginx-fdfs.conf

1.5.3 测试

通过 java 客户端上传文件 , 使用浏览器 http 访问文件 , 这里访问上传图片测试的文件 :

访问 storage :

http://192.168.101.5/group1/M00/00/00/wKhIBVVY2M-AM_9DAAAT7-0xdqM485_-



big.png

ip 地址改为 192.168.101.6 也可以访问到文件 ,因为同一个分组的 storage 文件互相同步。

访问 tracker : 访问 tracker 还需要 nginx 代为转发 , 可以使用模块 (需要安装集成) , 也可以自己在 nginx 中配置路由

http://192.168.101.3/group1/M00/00/00/wKhIBVVY2M-AM_9DAAAT7-0xdqM485_.big.png

big.png

ip 地址改为 192.168.101.4 也可以访问到文件。

使用域名访问 (推荐) :

nginx 对外由 vip 提供服务 , 使用域名访问如下 :

比如 vip 对应的域名为 img.test.com :

http://img.test.com/group1/M00/00/00/wKhIBVVY2M-AM_9DAAAT7-0xdqM485_big.png

big.png

这位写的不错

https://blog.csdn.net/qq_34898847/article/details/82258453

<https://blog.csdn.net/poehant/article/details/6996047>

https://www.jianshu.com/p/c7133383a4e7

问题
<https://www.cnblogs.com/xoray007/p/3640571.html>

不要胡乱修改 ip 和 store_path , 会出问题的 , store_path 不对应

