



分布式存储 CentOS6.5虚拟机环境搭建FastDFS-5.0.5集群 - OPEN 开发经验库

笔记本：技术资料

创建时间：2015/10/4 17:31

URL：<http://www.open-open.com/lib/view/open1435468300700.html>

分布式存储 CentOS6.5虚拟机环境搭建FastDFS-5.0.5集群

您的评价：

[收藏该经验](#)

前言:

由于公司项目需要，最近开始学习一下分布式存储相关知识，确定使用FastDFS这个开源工具。利用周末的时间在虚拟机上搭建了分布式存储系统，在搭建过程中，发现网上的资料说的并不是很全，有些细节需要单独搜索或者自己判断，所以现在想写这篇博客尽量记录一下自己在过程中遇到的问题，以及是如何解决的，方便自己也希望能帮忙到其他人。

=====

长长的分割线

=====

正文:

本次的目的是利用VMware搭建一个属于自己的hadoop集群。本次我们选择的是VMware10，具体的安装步骤大家可以到网上搜索，资源很多。

在阅读本篇文章之前，你最好对FastDFS有一个初步的了解，我建议可以看看下面的链接文章中的介绍:

<http://blog.csdn.net/poechant/article/details/6977407>

第一步，确定目标:

Tracker 192.168.224.20:22122 CentOS

Group1-Storage11 192.168.224.25:23000 CentOS

Group1-Storage12 192.168.224.26:23000 CentOS

Group2-Storage21 192.168.224.28:23001 CentOS

Group2-Storage22 192.168.224.29:23001 CentOS

虽然上述集群结构中tracker是单一节点，如果在生产环境中，这是不正确的，但现在我们主要是想通过这次的搭建熟悉基本的搭建流程，所以此处就暂时忽略吧。

第二步，确定安装版本：



 fastdfs-5.05.tar.gz	2015/5/10 星期...	好压 GZ 压缩文件	329 KB
 fastdfs-nginx-module_v1.16.tar.gz	2015/5/10 星期...	好压 GZ 压缩文件	18 KB
 libfastcommon-1.0.7.zip	2015/5/10 星期...	好压 ZIP 压缩文件	95 KB
 nginx-1.7.8.tar.gz	2015/5/10 星期...	好压 GZ 压缩文件	804 KB
 ngx_cache_purge-2.1.tar.gz	2015/5/11 星期...	好压 GZ 压缩文件	11 KB
 perl-5.20.2.tar.gz	2015/5/10 星期...	好压 GZ 压缩文件	11,696 KB

如上图可以看出，截止到2015年5月17日，fastdfs的最新版本是5.05，区别于以往版本，这个版本将公共的一些函数等单独封装成了libfastcommon这个jar包，所以在安装fastdfs之前，我们必须先安装libfastcommon。

同时，我们搭建完的分布式环境，还需要nginx帮忙实现http访问以及负载均衡、缓存等功能，涉及了nginx-1.7.8.tar.gz，fastdfs-nginx-module_v1.16.tar.gz，ngx_cache_purge-2.1.tar.gz。

PS:顺便分享一个网址，这个网址中有很多nginx的模块包，可查询下载: <http://labs.frickle.com/files/>

第三步，安装虚拟机环境：

如果没有搭建过虚拟机环境，可以参考我的另外两篇博客<http://www.cnblogs.com/PurpleDream/p/4263465.html>和<http://www.cnblogs.com/PurpleDream/p/426>主要讲解了如何安装一个CentOS6.5的虚拟机以及如何配置虚拟机的网络，还有如何根据已有的虚拟机，克隆多台虚拟机。

第四步，文件夹初始化:

按照我们第一步确认的目标，我们可以现在自由选择一个路径，创建一些文件夹目录，这些目录后边会在配置tracker和storage的过程中被用到。我在用虚拟机创建的时候，是在/opt这个目录下面创建相关的文件夹，详细如下，仅供参考:

1. 配置tracker所需的base_path: /opt/fastdfs_tracker。
2. 配置storage所需的日志目录: /opt/fastdfs_storage_info。备注: 这个目录是用来存储storage之间同步文件等日志的
3. 配置storage所需的存储文件目录: /opt/fastdfs_storage_data。备注: 这个目录是用来存储文件的

其实，我们在使用过程中并不是每个物理机上都会同时部署tracker和storage，那么根据自己的实际情况，如果只部署storage，则就创建storage的那两个文件夹；反之则只创建tracker的那个文件夹。后边在解释配置文件如何配置的过程中我将会按照上边的目录路径做配置，大家可以比对着看每个路径对应哪个配置。

第五步，安装libfastcommon-1.0.7.zip:

在安装libfastcommon的过程中，涉及了解压缩、make安装等过程，第一次安装的时候，可能由于环境的原因，会提示比如没有安装unzip、zip；没有安装perl；没有安装gcc等，逐一安装即可。

1. 利用unzip解压缩，如果提示没有安装，则按照下图中的命令安装即可：



```
[root@storage22 libfastcommon1.0.7]# unzip libfastcommon-1.0.7.zip
-bash: unzip: command not found
[root@storage22 libfastcommon1.0.7]# yum -y install unzip zip
Loaded plugins: fastestmirror
base                                     | 3.7 kB      00:00
base/primary_db                       | 3.6 MB      00:04
extras                                | 3.4 kB      00:00
extras/primary_db                    | 29 kB      00:00
updates                               | 3.4 kB      00:00
^[[^Aupdates/primary_db              16% [==          1 40 kB/s | 596 kB    01:17
updates/primary_db                   21% [===         1 46 kB/s | 783 kB    01:03 ETA
```

2. 解压完成后，进入文件夹，会看到make.sh文件，执行./make.sh命令，可能会报没有安装gcc和perl的错误，截图如下：

```
[root@storage22 libfastcommon-1.0.7]# ll
total 24
-rw-r--r-- 1 root root 2170 Sep 16 2014 HISTORY
-rw-r--r-- 1 root root 582 Sep 16 2014 INSTALL
-rw-r--r-- 1 root root 1341 Sep 16 2014 libfastcommon.spec
-rwxr-xr-x 1 root root 2151 Sep 16 2014 make.sh
-rw-r--r-- 1 root root 617 Sep 16 2014 README
drwxr-xr-x 2 root root 4096 Sep 16 2014 src
[root@storage22 libfastcommon-1.0.7]# ./make.sh
./make.sh: line 14: gcc: command not found
./make.sh: line 15: ./a.out: No such file or directory
./make.sh: line 99: perl: command not found
./make.sh: line 100: perl: command not found
cc -c -fPIC -o hash.lo hash.c
make: cc: Command not found
make: *** [hash.lo] Error 127
[root@storage22 libfastcommon-1.0.7]# _
```

3. 为了解决2中的问题，安装gcc和perl：

a. 执行 `yum -y install gcc` 和 `yum -y install gcc-c++` 这两个命令安装

b. 安装perl时，我们可以直接用wget命令安装，我这里安装的版本是perl-5.20.2。

可以参考这篇文章安装perl http://blog.sina.com.cn/s/blog_68158ebf0100ndes.html

wget <http://www.cpan.org/src/5.0/perl-5.20.2.tar.gz>

tar zxvf perl-5.20.2.tar.gz

mkdir /usr/local/perl

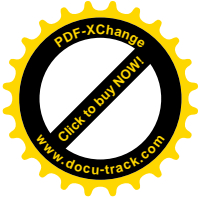
./Configure -des -Dprefix=/usr/local/perl -Dusethreads -Uversiononly

make

make install

perl -version

4. 安装完perl后，重新依次执行 `./make.sh` 和 `./make.sh install` 这两个命令，完成libfastcommon的安装。



```
-rw-rw-r-- 1 root root 2802 Nov 22 2014 fastdfs.spec
-rw-rw-r-- 1 root root 31386 Nov 22 2014 HISTORY
drwxrwxr-x 2 root root 4096 Nov 22 2014 init.d
-rw-rw-r-- 1 root root 7755 Nov 22 2014 INSTALL
-rwxrwxr-x 1 root root 5813 Nov 22 2014 make.sh
drwxrwxr-x 2 root root 4096 Nov 22 2014 php_client
-rw-rw-r-- 1 root root 2380 Nov 22 2014 README.md
-rwxrwxr-x 1 root root 1768 Nov 22 2014 restart.sh
-rwxrwxr-x 1 root root 1680 Nov 22 2014 stop.sh
drwxrwxr-x 4 root root 4096 Jun 27 21:35 storage
drwxrwxr-x 2 root root 4096 Nov 22 2014 test
drwxrwxr-x 2 root root 4096 Jun 27 21:35 tracker
[root@storage22 fastdfs-5.05]# cd /etc/fdfs/
[root@storage22 fdfs]# ll
total 20
-rw-r--r-- 1 root root 1461 Jun 27 21:35 client.conf.sample
-rw-r--r-- 1 root root 7829 Jun 27 21:35 storage.conf.sample
-rw-r--r-- 1 root root 7102 Jun 27 21:35 tracker.conf.sample
[root@storage22 fdfs]#
[root@storage22 fdfs]#
[root@storage22 fdfs]#
[root@storage22 fdfs]#
[root@storage22 fdfs]#
[root@storage22 fdfs]#
```



第七步，配置tracker:

前边的这六步无论是配置tracker还是配置storage都是必须的，而tracker和storage的区别主要是在安装完fastdfs之后的配置过程中。我们在第一步确认目标时，计划在192.168.224.20这台虚拟机上配置tracker，所以我们就在这台机器上演示。

1. 进入/etc/fdfs文件夹，执行命令: cp tracker.conf.sample tracker.conf。

2. 编辑tracker.conf，执行命令: vi tracker.conf，将以下几个选项进行编辑:

a. disabled=false #启用配置文件

b. port=22122 #设置tracker的端口号，一般采用22122这个默认端口

c. base_path=/opt/fastdfs_tracker #设置tracker的数据文件和日志目录（预先创建）

d. http.server_port=8080 #设置http端口号 注意，这个配置在fastdfs5.05这个版本中已经不用配置，不用管这个！

```
# is this config file disabled
# false for enabled
# true for disabled
disabled=false

# bind an address of this host
# empty for bind all addresses of this host
bind_addr=

# the tracker server port
port=22122

# connect timeout in seconds
# default value is 30s
connect_timeout=30

# network timeout in seconds
# default value is 30s
network_timeout=60

# the base path to store data and log files
base_path=/opt/fastdfs_tracker

# max concurrent connections this server supported
--More-- (6%)
```

3. 启动tracker，执行如下命令: /usr/local/bin/fdfs_trackerd /etc/fdfs/tracker.conf restart

注意，一般fdfs_trackerd等命令在/usr/local/bin中没有，而是在/usr/bin路径下，所以命令修改如下: /usr/bin/fdfs_trackerd /etc/fdfs/tracker.conf restart

4. 启动完毕后，可以通过以下两个方法查看tracker是否启动成功:

a. netstat -unltp|grep fdfs，查看22122端口监听情况

b. 通过以下命令查看tracker的启动日志，看是否有错误: tail -100f /opt/fastdfs_tracker/logs/trackerd.log

5. 如果启动没有问题，可以通过以下步骤，将tracker的启动添加到服务器的开机启动中:

a. 打开文件 vi /etc/rc.d/rc.local

b. 将如下命令添加到该文件中 /usr/bin/fdfs_trackerd /etc/fdfs/tracker.conf restart

第八步，配置storage:

其实配置storage和配置tracker类似，只不过配置文件和配置内容不一样。我们以配置192.168.224.29配置storage为例。



1. 进入/etc/fdfs文件夹，执行命令: cp storage.conf.sample storage.conf。
2. 编辑storage.conf，执行命令: vi storage.conf，将以下几个选项进行编辑:
 - a. disabled=false #启用配置文件
 - b. group_name=group2 #组名，根据实际情况修改
 - c. port=23001 #设置storage的端口号，默认是23000，同一个组的storage端口号必须一致
 - d. base_path=/opt/fastdfs_storage_info #设置storage的日志目录（需预先创建）
 - e. store_path_count=1 #存储路径个数，需要和store_path个数匹配
 - f. store_path0=/opt/fastdfs_storage_data #存储路径
 - g. tracker_server=192.168.224.20:22122 #tracker服务器的IP地址和端口号
 - h. http.server_port=8080 #设置http端口号 注意，这个配置在fastdfs5.05这个版本中已经不用配置，不用管这个！

```
# is this config file disabled
# false for enabled
# true for disabled
disabled=false

# the name of the group this storage server belongs to
#
# comment or remove this item for fetching from tracker server,
# in this case, use_storage_id must set to true in tracker.conf,
# and storage_ids.conf must be configured correctly.
group_name=group2

# bind an address of this host
# empty for bind all addresses of this host
bind_addr=

# if bind an address of this host when connect to other servers
# (this storage server as a client)
# true for binding the address configed by above parameter: "bind_addr"
# false for binding any address of this host
client_bind=true

# the storage server port
port=23001
```

```
# heart beat interval in seconds
heart_beat_interval=30

# disk usage report interval in seconds
stat_report_interval=60

# the base path to store data and log files
base_path=/opt/fastdfs_storage_info

# max concurrent connections the server supported
# default value is 256
# more max_connections means more memory will be used
max_connections=256

# the buff size to recv / send data
# this parameter must more than 8KB
# default value is 64KB
# since 02.00
buff_size = 256KB

# accept thread count
# default value is 1
# since 04.07
accept_threads=1
```




```
# write to the mark file after sync N files
# default value is 500
write_mark_file_freq=500

# path(disk or mount point) count, default value is 1
store_path_count=1

# store_path#, based 0, if store_path0 not exists, it's value is base_path
# the paths must be exist
store_path0=/opt/fastdfs_storage_data
store_path1=/home/yuqing/fastdfs2

# subdir_count * subdir_count directories will be auto created under each
# store_path (disk), value can be 1 to 256, default value is 256
subdir_count_per_path=256

# tracker_server can occur more than once, and tracker_server format is
# "host:port", host can be hostname or ip address
tracker_server=192.168.224.20:22122

#standard log level as syslog, case insensitive, value list:
### emerg for emergency
### alert
### crit for critical
```

3. 启动storage，执行如下命令: /usr/local/bin/fdfs_storage /etc/fdfs/storage.conf restart

注意，一般fdfs_storage等命令在/usr/local/bin中没有，而是在/usr/bin路径下，所以命令修改如下: /usr/bin/fdfs_storage /etc/fdfs/storage.conf restart

4. 启动完毕后，可以通过以下两个方法查看storage是否启动成功:

a. netstat -unltp|grep fdfs，查看23001端口监听情况

b. 通过以下命令查看storage的启动日志，看是否有错误: tail -100f

/opt/fastdfs_storage_info/logs/storage.log

5. 启动成功后，可以通过fdfs_monitor查看集群的情况，即storage是否已经注册到tracker服务器中

/usr/bin/fdfs_monitor /etc/fdfs/storage.conf

查看192.168.224.29:23001 是ACTIVE状态即可

6. 如果启动没有问题，可以通过以下步骤，将storage的启动添加到服务器的开机启动中:

a. 打开文件 vi /etc/rc.d/rc.local

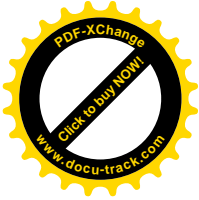
b. 将如下命令添加到该文件中 /usr/bin/fdfs_storage /etc/fdfs/storage.conf restart

=====
长长的分割线
=====

以上八步其实已经完成了fastdfs的配置，如果此时你用java等api编辑客户端，实际上就可以完成文件的上传、同步和下载。但是为什么网上还会有很多人说需要nginx呢？？其实主要原因时因为，我们可以通过配置nginx为下载提供基于http协议的下载等功能。

其实，storage中安装nginx，主要是为了为提供http的访问服务，同时解决group中storage服务器的同步延迟问题。而tracker中安装nginx，主要是为了提供http访问的反向代理、负载均衡以及缓存服务

第九步，安装nginx的准备:



不管是在tracker中还是storage中安装nginx，前提都需要安装一些基础软件。一些大公司的服务器默认都会初始化这些软件，但是你在配置的时候可能还是最好自己利用命令确认一下。

1. yum install -y gcc 这个前边在安装libfastcommon之前已经安装了
2. yum install -y gcc-c++ 这个前边在安装libfastcommon之前已经安装了
3. yum install -y pcre pcre-devel
4. yum install -y zlib zlib-devel
5. yum install -y openssl openssl-devel

第十步，在storage中安装nginx:

1. 创建nginx默认的安装文件夹: mkdir /usr/local/nginx
2. 提前将 nginx1.7.8.tar.gz 和 fastdfs-nginx-module_v1.16.tar.gz 解压缩，然后进入nginx1.7.8的文件夹目录，执行如下命令:

```
./configure --prefix=/usr/local/nginx --add-module=/myself_settings/fastdfs-nginx-module/fastdfs-nginx-module/src
```

3. 执行完上述命令，如果没有报错的话，我们继续执行 make 命令，此时编译有可能会报错，那是因为我们忘了做一项重要的工作，我们可以参考下面这篇文章<http://bbs.chinaunix.net/thread-4163021-1-1.html>中的解答，其实汇总就是下面两个意思:

a. 我们在解压缩fastdfs-nginx-module_v1.16.tar.gz之后，需要进入fastdfs-nginx-module/src目录，编辑config文件，找到包含CORE_INCS这一行，将路径中local全部去掉，变为CORE_INCS="\$CORE_INCS /usr/include/fastdfs /usr/include/fastcommon/"

b. 建立软连接，这个软连接我们在第五步安装libfastcommon时的最后一个操作就已经建立了，所以此处不用再建立了。

4. 执行完3个步骤后，我们在重复执行2中的命令，然后再依次执行 make 和 make install 这两个命令，没有报错误就是安装成功了。

5. 执行命令 cd /usr/local/nginx/conf，编辑 nginx.conf 这个文件，编辑如下:

```
listen    8080;
```

在server段中添加：

```
location ~/group[1-3]/M00{
    root/fdfs/storage/data;

    ngx_fastdfs_module;
}
```

注意，如果配置的storage是在group2组，则下面的location应该是 ~/group2/M00



```
#gzip on;

server {
    listen 8080;
    server_name localhost;

    #charset koi8-r;

    #access_log logs/host.access.log main;

    location ~ /group2/M00 {
        root /opt/fastdfs_storage_data/data;
        ngx_fastdfs_module;
    }

    location / {
        root html;
        index index.html index.htm;
    }

    #error_page 404 /404.html;

    # redirect server error pages to the static page /50x.html
```

6. 执行命令 `cd /myself_settings/fastdfs5.0.5/fastdfs-5.05/conf`，即进入fastdfs5.0.5的安装文件夹的conf目录下，**将目录下面的http.conf和mime.types拷贝到/etc/fdfs/下，如果不执行这一步，后边在启动nginx时会报错。**

7. 执行命令 `cd /myself_settings/fastdfs_nginx_module/fastdfs-nginx-module/src`，即进入 fastdfs-nginx-module_v1.16的安装文件夹的src目录下，将目录下面的mod_fastdfs.conf这个文件拷贝到 /etc/fdfs 目录下。

8. 打开 /etc/fdfs 这个目录，编辑 mod_fastdfs.conf 这个文件，如下：

注意以下配置是group2的第二个storage 192.168.224.29的配置，按照第一步确定的集群目标来配置的，供大家参考。

- a. `base_path=/opt/fastdfs_storage_info` #保存日志目录
- b. `tracker_server=192.168.224.20:22122` #tracker服务器的IP地址以及端口号
- c. `storage_server_port=23001` #storage服务器的端口号
- d. `group_name=group2` #当前服务器的group名
- e. `url_have_group_name= true` #文件url中是否有group名
- f. `store_path_count=1` #存储路径个数，需要和store_path个数匹配
- g. `store_path0=/opt/fastdfs_storage_data` #存储路径
- f. `http_need_find_content_type=true` #从文件扩展名查找文件类型（nginx时为true） **注意:这个配置网上一般都会列出，但是在fastdfs5.05的版本中是没有的**
- h. `group_count= 2` #设置组的个数
- i. 在文件的末尾，按照第一步确定的集群目标，追加如下图的配置：



```
# since v1.14
group_count = 2

# group settings for group #1
# since v1.14
# when support multi-group, uncomment following section
[group1]
group_name=group1
storage_server_port=23000
store_path_count=2
store_path0=/opt/fastdfs_storage_data
store_path1=/opt/fastdfs_storage_data

# group settings for group #2
# since v1.14
# when support multi-group, uncomment following section as neccessary
[group2]
group_name=group2
storage_server_port=23001
store_path_count=2
store_path0=/opt/fastdfs_storage_data
store_path1=/opt/fastdfs_storage_data
```

9. 建立软连接 `ln -s /opt/fastdfs_storage_data/data /opt/fastdfs_storage_data/data/M00`
10. 执行命令启动nginx: `/usr/local/nginx/sbin/nginx` , 如果 `/usr/local/nginx/logs/error.log` 中没有报错, 同时访问 `192.168.224.29:8080` 这个url能看到 nginx的欢迎页面。

第十一步, 在tracker中安装nginx:

注意, tracker中nginx安装时比storage中的nginx安装时多安装一个nginx的缓存模块, 同时在配置的时候有很大的不同。

1. 创建nginx默认的安装文件夹: `mkdir /usr/local/nginx`
2. 提前将 `nginx1.7.8.tar.gz`、`fastdfs-nginx-module_v1.16.tar.gz`、`ngx_cache_purge-2.1.tar.gz` 解压缩, 然后进入 `nginx1.7.8` 的文件夹目录, 执行如下命令:

```
./configure --prefix=/usr/local/nginx --add-module=/myself_settings/fastdfs-nginx-  
module/fastdfs-nginx-module/src --add-module=/myself_settings/ngx_cache_purge/ngx_cache_purge-  
2.1
```

3. 执行完上述命令, 如果没有报错的话, 我们继续执行 `make` 命令, 此时编译有可能会报错, 那是因为我们忘了做一项重要的工作, 我们可以参考下面这篇文章<http://bbs.chinaunix.net/thread-4163021-1-1.html> 中的解答, 其实汇总就是下面两个意思:

a. 我们在解压缩 `fastdfs-nginx-module_v1.16.tar.gz` 之后, 需要进入 `fastdfs-nginx-module/src` 目录, 编辑 `config` 文件, 找到包含 `CORE_INCS` 这一行, 将路径中 `local` 全部去掉, 变为 `CORE_INCS="$CORE_INCS /usr/include/fastdfs /usr/include/fastcommon/"`

b. 建立软连接, 这个软连接我们在第五步安装 `libfastcommon` 时的最后一个操作就已经建立了, 所以此处不用再建立了。

4. 执行完3个步骤后, 我们在重复执行2中的命令, 然后再依次执行 `make` 和 `make install` 这两个命令, 没有报错误就是安装成功了。

5. 执行命令 `cd /usr/local/nginx/conf`, 编辑 `nginx.conf` 这个文件, 编辑如下:

```
1 | #user nobody;  
2 | worker_processes 1;  
3 |
```





```
4 #error_log logs/error.log;
5 #error_log logs/error.log notice;
6 #error_log logs/error.log info;
7
8 #pid logs/nginx.pid;
9
10
11 events {
12     worker_connections 1024;
13 }
14
15
16 http {
17     include mime.types;
18     default_type application/octet-stream;
19
20     #log_format main '$remote_addr - $remote_user [$time_local] "$request" '
21     # '$status $body_bytes_sent "$http_referer" '
22     # '"$http_user_agent" "$http_x_forwarded_for"';
23
24     #access_log logs/access.log main;
25
26     sendfile on;
27     tcp_nopush on;
28
29     #keepalive_timeout 0;
30     keepalive_timeout 65;
31
32     #gzip on;
33
34     server_names_hash_bucket_size 128;
35     client_header_buffer_size 32k;
36     large_client_header_buffers 4 32k;
37
38     client_max_body_size 300m;
39
40     proxy_redirect off;
41     proxy_set_header Host $http_host;
42     proxy_set_header X-Real-IP $remote_addr;
43     proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
44
45     proxy_connect_timeout 90;
46     proxy_send_timeout 90;
47     proxy_read_timeout 90;
48
49     proxy_buffer_size 16k;
50     proxy_buffers 4 64k;
51     proxy_busy_buffers_size 128k;
52     proxy_temp_file_write_size 128k;
53
54     proxy_cache_path /opt/cache/nginx/proxy_cache levels=1:2
55     keys_zone=http-cache:500m max_size=10g inactive=30d;
56     proxy_temp_path /opt/cache/nginx/proxy_cache/tmp;
57
58     upstream fdfs_group1 {
59         server 192.168.224.25:8080 weight=1 max_fails=2 fail_timeout=30s;
60         server 192.168.224.26:8080 weight=1 max_fails=2 fail_timeout=30s;
61     }
62
63     upstream fdfs_group2 {
64         server 192.168.224.28:8080 weight=1 max_fails=2 fail_timeout=30s;
65         server 192.168.224.29:8080 weight=1 max_fails=2 fail_timeout=30s;
66     }
67
68     server {
69         listen 8080;
70         server_name localhost;
71
72         #charset koi8-r;
73
74         #access_log logs/host.access.log main;
75
76         location /group1/M00 {
77             proxy_next_upstream http_502 http_504 error timeout invalid_header
78             proxy_cache http-cache;
```



```
79         proxy_cache_valid 200 304 12h;
80         proxy_cache_key $uri$is_args$args;
81         proxy_pass http://fdfs_group1;
82         expires 30d;
83     }
84
85     location /group2/M00 {
86         proxy_next_upstream http_502 http_504 error timeout invalid_header
87         proxy_cache http-cache;
88         proxy_cache_valid 200 304 12h;
89         proxy_cache_key $uri$is_args$args;
90         proxy_pass http://fdfs_group2;
91         expires 30d;
92     }
93
94     location ~ /purge(/.*) {
95         allow 127.0.0.1;
96         allow 192.168.224.0/24;
97         deny all;
98         proxy_cache_purge http-cache $1$is_args$args;
99     }
100
101     #error_page 404 /404.html;
102
103     # redirect server error pages to the static page /50x.html
104     #
105     error_page 500 502 503 504 /50x.html;
106     location = /50x.html {
107         root html;
108     }
109
110     # proxy the PHP scripts to Apache listening on 127.0.0.1:80
111     #
112     #location ~ /\.php$ {
113     #    proxy_pass http://127.0.0.1;
114     #}
115
116     # pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
117     #
118     #location ~ /\.php$ {
119     #    root html;
120     #    fastcgi_pass 127.0.0.1:9000;
121     #    fastcgi_index index.php;
122     #    fastcgi_param SCRIPT_FILENAME /scripts$fastcgi_script_name;
123     #    include fastcgi_params;
124     #}
125
126     # deny access to .htaccess files, if Apache's document root
127     # concurs with nginx's one
128     #
129     #location ~ /\.ht {
130     #    deny all;
131     #}
132 }
133
134 # another virtual host using mix of IP-, name-, and port-based configuration
135 #
136 #server {
137 #    listen 8000;
138 #    listen somename:8080;
139 #    server_name somename alias another.alias;
140
141     #    location / {
142     #        root html;
143     #        index index.html index.htm;
144     #    }
145     #}
146
147
148 # HTTPS server
149 #
150 #server {
151 #    listen 443 ssl;
152 #    server_name localhost;
153
```



```
154 # ssl_certificate cert.pem;
155 # ssl_certificate_key cert.key;
156
157 # ssl_session_cache shared:SSL:1m;
158 # ssl_session_timeout 5m;
159
160 # ssl_ciphers HIGH:!aNULL:!MD5;
161 # ssl_prefer_server_ciphers on;
162
163 # location / {
164 #     root html;
165 #     index index.html index.htm;
166 # }
167 #}
```

6. 执行命令启动nginx: `/usr/local/nginx/sbin/nginx` , 如果`/usr/local/nginx/logs/error.log`中没有报错, 同时访问`192.168.224.20:8080`这个url能看到 nginx的欢迎页面。

第十二步, 测试上传:

1. 打开 `/etc/fdfs` 文件夹, 编辑 `client.conf` 文件, 编辑内容如下:

a. `base_path=/opt/fastdfs_tracker` #存放路径

b. `tracker_server=192.168.224.20:22122` #tracker服务器IP地址和端口号

c. `http.tracker_server_port=8080` #tracker服务器的http端口号, 注意, 这个配置在fastdfs5.0.5

中已经没有了

2. 模拟上传文件, 执行如下命令: `/usr/bin/fdfs_upload_file /etc/fdfs/client.conf /opt/1.txt`

使用浏览器访问返回的

url: `http://192.168.224.20:8080/group1/M00/00/00/wKjgGIVYgi6AAv3tAAADv4ZzcQ572.txt`

也可以直接访问文件所在的

storage: `http://192.168.224.25:8080/group1/M00/00/00/wKjgGIVYgi6AAv3tAAADv4ZzcQ572.txt`

上述如果访问成功, 会在 tracker 192.168.224.20 的 `/opt/cache`中产生缓存

我们还可以通过在url中添加purge清除缓存, 例

如: `http://192.168.224.20:8080/purge/group1/M00/00/00/wKjgGIVYgi6AAv3tAAADv4ZzcQ572.txt`

参考资料:

<http://blog.csdn.net/lynnlovemin/article/details/39398043> fastdfs集群的配置教程

<http://blog.csdn.net/poechant/article/details/6977407> fastdfs系列教程

<http://m.blog.csdn.net/blog/hfty290/42030339> tracker-leader的选举



相关资讯 — 更多

[Linux运维工程师的十个基本技能点](#)
[主流开源云计算软件大盘点](#)
[XenServer 6.5 正式发布，期盼已久，不负所望！](#)
[ZStack v0.8 发布，开源 IaaS 软件](#)
[Docker周报第17期](#)
[Linux 运维工程师的十个基本技能点](#)
[入门Linux运维工程师，必须要掌握的10个技术点](#)
[推荐 8 个开源的 Linux 防火墙](#)
[OpenStack在小米私有云平台的实践](#)
[基于开源软件在Azure平台建立大规模系统的最佳实践](#)
[国外程序员整理的系统管理员资源大全](#)
[如何在 CentOS 7 上安装 Docker](#)
[初创公司利用Docker持续集成部署](#)
[CoreOS那些事之系统升级](#)
[关于 Apache 的 25 个初中级面试题](#)
[CentOS 6.2 发布下载](#)
[深入理解docker ulimit](#)
[项目管理 - 禅道 2.2 版本发布，内置 SVN 集成功能](#)
[Oracle Linux 真的是 CentOS 的更好替代品吗？](#)
[应该知道的Linux技巧](#)

相关文档 — 更多

[Debian+ Nginx + PHP\(FastCGI\)+ MySQL搭建LNMP服务器.debianlnmp.doc](#)
[Oracle数据库自动备份手册\(Linux环境\).doc](#)
[Centos6 SSH 配置.docx](#)
[01-xen安装-赶星老师视频资料.pdf](#)
[在CentOS 6.x通过RPM包安装OpenStack多节点环境.pdf](#)
[CentOS6怎么样设置ADSL上网.doc](#)
[Hadoop现场演示与编程过程.ppt](#)
[Centos6/Installation/Minimal installation using yum.pdf](#)
[Lxc/Installation/Guest/Centos/6.pdf](#)
[Centos 6 Linux Server Cookbook.pdf](#)
[CentOS 6 Linux Server Cookbook.pdf](#)
[Centos5安装注意文档.doc](#)
[EasyHadoop 实战手册.doc](#)
[在CentOS 5系统上安装ffmpeg详细的方法.doc](#)
[CentOS 5 下安装 tfs 并使用 Nginx 做访问.docx](#)
[Centos6 Hadoop 安装笔记.docx](#)
[CentOS 6.3安装配置LAMP服务器 \(Apache+ PHP5 +MySQL\).pdf](#)
[CentOS5 命令.doc](#)
[CentOS 6 Linux Server Cookbook.pdf](#)
[Zabbix中文使用手册.pdf](#)

