# FastDFS 部署

原创 酥心糖 2018-04-02 18:37:00 评论(0) 125人阅读

### FastDFS 介绍

FastDFS是一个开源的轻量级分布式文件系统，它对文件进行管理，功能包括：文件存储、文件同步、文件访问（文件上传、文件下载）等，解决了大容量存储和负载均衡的问题。特别适合中小文件（建议范围：4KB < file_size <500MB）以文件为载体的在线服务，如相册网站、视频网站等。
FastDFS服务端有两个角色：跟踪器（tracker）和存储节点（storage）。跟踪器主要做调度工作，在访问上起负载均衡的作用。

跟踪器（tracker）：负责管理所有的 storage server和 group，每个 storage 在启动后会连接 Tracker，告知自己所属 group 等信息，并保持周期性心跳。tracker非常容易扩展，直接增加tracker机器即可扩展为tracker cluster。
存储节点（storage）： 主要提供容量和备份服务；以 group 为单位，每个 group 内可以有多台 storage server，数据互为备份。
Group ：组， 也可称为卷。 相同组内服务器上的文件是完全相同的 ，同一组内的storage server之间是对等的， 文件上传、 删除等操作可以在任意一台storage server上进行 。不同组之间存放的文件不同，互为分布式存取。

fastdfs原理请参考： http://blog.chinaunix.net/uid-20196318-id-4058561.html

### 环境准备

两台Tracker主机：
tracker： 192.168.0.1
tracker：192.168.0.2
四台Storage主机：
[group1]
sortage1: 192.168.0.3
sortage2: 192.168.0.4

[group2]
sortage3: 192.168.0.5
sortage4: 192.168.0.6
软件包：

wget https://github.com/happyfish100/libfastcommon/archive/master.zip

fastDFS软件包：http://sourceforge.net/projects/fastdfs/files/

storage ngnx扩展包：https://sourceforge.net/projects/fastdfs/files/FastDFS%20Nginx%20Module%20Source%20Code/

Tracker nginx扩展包：

http://labs.frickle.com/nginx_ngx_cache_purge/

## 安装FastDFS-Tracker

1、所有节点安装环境依赖包：

```
yum install make cmake gcc gcc-c++ pcre pcre-devel zlib zlib-devel openssl openssl-d
wget https://github.com/happyfish100/libfastcommon/archive/master.zip
unzip master.zip
cd libfastcommon-master/
./make.sh
./make.sh  install
```

2、所有节点安装FastDFS：

```
tar xf FastDFS_v5.08.tar.gz
cd FastDFS
./make.sh
./make.sh install
cd /etc/fdfs/
```

## 配置Tracker Server

1、将示例文件修改为配置文件

```
cd /etc/fdfs/
cp tracker.conf.sample  tracker.conf
```

2、编辑配置文件，指定如下参数,其它参数可以根据自己的实际需求修改

```
bind_addr=192.168.0.1   # 本机IP地址
port=22122                              # 监听的端口
base_path=/fastdfs/tracker    # 数据目录
```

具体的配置参数介绍可以参考：http://bbs.chinaunix.net/thread-1941456-1-1.html

3、创建程序目录

```
mkdir  -p /fastdfs/tracker
```

4、启动Tracker Server (在CentOS7的版本中，也使用此命令启动，没有systemd的管理配置)

```
/etc/init.d/fdfs_trackerd start
```

5、查看端口和服务是否启动：

```
# netstat -lntp|grep 22122
tcp        0      0 192.168.0.1:22122       0.0.0.0:*               LISTEN      20315/fdf
```

### 安装FasterDFS-Storage

> 提示： 所有storage节点执行如下配置。

1、编辑storage1和 sortage2主机上的配置文件：

```
cd /etc/fdfs/
cp storage.conf.sample storage.conf
vim storage.conf:

group_name=group1
base_path=/fastdfs/storage
store_path_count=1
store_path0=/fastdfs/storage
tracker_server=192.168.0.1:22122
tracker_server=192.168.0.2:22122
http.server_port=8888
```

编辑 storage3和storage4主机上的配置：

```
cd /etc/fdfs/
cp storage.conf.sample storage.conf
vim storage.conf:

group_name=group2
base_path=/fastdfs/storage
store_path_count=1
store_path0=/fastdfs/storage
tracker_server=192.168.0.1:22122
tracker_server=192.168.0.2:22122
http.server_port=8888
```

2、启动服务，并设置开机自启动：

```
/etc/init.d/fdfs_storaged  start
chkconfig --add fdfs_storaged
```

3、查看端口和服务：

```
# netstat -lntp|grep 23000
tcp        0      0 192.168.0.3:23000       0.0.0.0:*               LISTEN      20772/fdf
```

### Storage & Nginx 配置

> 提示： 如果没有特殊说明，需要在所有storage服务器上进行操作。

1、上传nginx的扩展包，解压：

```
tar xf fastdfs-nginx-module_v1.16.tar.gz
cd fastdfs-nginx-module/src
```

2、编辑文件，这里的路径需要修改，删掉 `local` 这一层，否则编译时会显示找不到文件：

```
vim config
#将源内容：
ngx_addon_name=ngx_http_fastdfs_module
HTTP_MODULES="$HTTP_MODULES ngx_http_fastdfs_module"
NGX_ADDON_SRCS="$NGX_ADDON_SRCS $ngx_addon_dir/ngx_http_fastdfs_module.c"
CORE_INCS="$CORE_INCS /usr/local/include/fastdfs /usr/local/include/fastcommon/"
CORE_LIBS="$CORE_LIBS -L/usr/local/lib -lfastcommon -lfdfsclient"
CFLAGS="$CFLAGS -D_FILE_OFFSET_BITS=64 -DFDFS_OUTPUT_CHUNK_SIZE='256*1024'
-DFDFS_MOD_CONF_FILENAME='\"/etc/fdfs/mod_fastdfs.conf\"'"

# 修改为：
ngx_addon_name=ngx_http_fastdfs_module
HTTP_MODULES="$HTTP_MODULES ngx_http_fastdfs_module"
NGX_ADDON_SRCS="$NGX_ADDON_SRCS $ngx_addon_dir/ngx_http_fastdfs_module.c"
CORE_INCS="$CORE_INCS /usr/include/fastdfs /usr/include/fastcommon/"
CORE_LIBS="$CORE_LIBS -L/usr/local/lib -lfastcommon -lfdfsclient"
CFLAGS="$CFLAGS -D_FILE_OFFSET_BITS=64 -DFDFS_OUTPUT_CHUNK_SIZE='256*1024' -DFDFS_
```

3、安装nginx扩展,先把之前的nginx删除：

```
yum remove nginx -y
```

4、上传nginx源码包，添加扩展模块进行编译：

```
./configure --prefix=/usr/local/nginx --add-module=/tmp/fastdfs-nginx-module/src/
make && make install
```

5、复制配置文件到/etc/fdfs目录:

```
cp /tmp/fastdfs-nginx-module/src/mod_fastdfs.conf /etc/fdfs/
```

6、storage1和strage2主机上修改配置文件：

```
vim  /etc/fdfs/mod_fastdfs.conf

#修改内容：
base_path= /fastdfs/storage
tracker_server=192.168.0.1:22122
tracker_server=192.168.0.2:22122
group_name= group1
url_have_group_name = true
store_path_count=1
store_path0=/fastdfs/storage
group_count= 2

#结尾处增加内容:
[group1]
groupname=group1
storage_server_port=23000
store_path_count=1
storepath0=/fastdfs/storage
[group2]
groupname=group2
storage_server_port=23000
store_path_count=1
store_path0=/fastdfs/storage
```

在storage3和storage4上执行如下操作：

```
vim  /etc/fdfs/mod_fastdfs.conf

#修改内容：
base_path= /fastdfs/storage
tracker_server=192.168.0.1:22122
tracker_server=192.168.0.2:22122
group_name= group2
url_have_group_name = true
store_path_count=1
store_path0=/fastdfs/storage
group_count= 2
```

```
# 结尾处增加内容:
[group1]
groupname=group1
storage_server_port=23000
store_path_count=1
storepath0=/fastdfs/storage
[group2]
groupname=group2
storage_server_port=23000
store_path_count=1
store_path0=/fastdfs/storage
```

7、修改nginx配置文件:

```
vim /usr/local/nginx/conf/nginx.conf
# 配置一个server

    server {
            listen       8888;
            server_name  localhost;
            location ~/group([0-9])/M00 {
        #alias /fastdfs/storage/data;
        #ngx_fastdfs_module;
    }
```

8、配置nginx

```
```
cp /usr/local/nginx/sbin/nginx /usr/sbin/
ln -s /fastdfs/storage/data/ /fastdfs/storage/data/M00
```
```

### Tracker & Nginx配置

> 提示：两台tracker配置相同

1、Tracker 上配置nginx需要安装一个缓存模块：http://labs.frickle.com/nginx_ngx_cache_purge/

2、先删除nginx:

```
yum remove nginx -y
```

3、解压文件，编译安装:

```
tar xf ngx_cache_purge-2.3.tar.gz
tar xf nginx-1.11.4.tar.gz
cd nginx-1.11.4
./configure --prefix=/usr/local/nginx --add-module=/tmp/ngx_cache_purge-2.3/
make && make install
```

4、修改nginx的配置文件

```
vim  /usr/llocal/nginx/conf/nginx.conf

#user  nobody;
worker_processes  auto;

#error_log  logs/error.log;
#error_log  logs/error.log  notice;
#error_log  logs/error.log  info;

#pid        logs/nginx.pid;

events {
    worker_connections  1024;
}

http {
    include       mime.types;
    default_type  application/octet-stream;

    #log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
    #                  '$status $body_bytes_sent "$http_referer" '
    #                  '"$http_user_agent" "$http_x_forwarded_for"';

    #access_log  logs/access.log  main;

    sendfile        on;
    tcp_nopush      on;

    #keepalive_timeout  0;
    keepalive_timeout  65;

    #gzip  on;

    server_names_hash_bucket_size  128;
    client_header_buffer_size  32k;
    large_client_header_buffers  4  32k;

    client_max_body_size  300m;

    proxy_redirect  off;
    proxy_set_header Host  $http_host;
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;

    proxy_connect_timeout 90;
    proxy_send_timeout 90;
    proxy_read_timeout 90;

    proxy_buffer_size 16k;
    proxy_buffers 4 64k;
    proxy_busy_buffers_size 128k;
    proxy_temp_file_write_size 128k;

    proxy_cache_path /fastdfs/cache/nginx/proxy_cache levels=1:2
    keys_zone=http-cache:500m max_size=10g inactive=30d;
    proxy_temp_path /fastdfs/cache/nginx/proxy_cache/tmp;

    upstream fdfs_group1 {
         server 192.168.60.3:8888 weight=1 max_fails=2 fail_timeout=30s;
         server 192.168.60.4:8888 weight=1 max_fails=2 fail_timeout=30s;
    }

    upstream fdfs_group2 {
         server 192.168.60.5:8888 weight=1 max_fails=2 fail_timeout=30s;
         server 192.168.60.6:8888 weight=1 max_fails=2 fail_timeout=30s;
    }

    server {
        listen       80;
        server_name  localhost;

        #charset koi8-r;

        #access_log  logs/host.access.log  main;

        location /group1/M00 {
            proxy_next_upstream http_502 http_504 error timeout invalid_header;
            proxy_cache http-cache;
            proxy_cache_valid  200 304 12h;
            proxy_cache_key $uri$is_args$args;
            proxy_pass http://fdfs_group1;
            expires 30d;
        }

        location /group2/M00 {
            proxy_next_upstream http_502 http_504 error timeout invalid_header;
            proxy_cache http-cache;
            proxy_cache_valid  200 304 12h;
            proxy_cache_key $uri$is_args$args;
            proxy_pass http://fdfs_group2;
            expires 30d;
        }
        location ~/purge(/.*) {
            allow 127.0.0.1;
            allow 192.168.0.0/24;
            deny all;
            proxy_cache_purge http-cache  $1$is_args$args;
        }

        location /clear {
             proxy_pass    http://127.0.0.1:8182;
             proxy_redirect default;
        }
        #error_page  404              /404.html;

        # redirect server error pages to the static page /50x.html
        #
        error_page   500 502 503 504  /50x.html;
        location = /50x.html {
            root   html;
        }
    }
}
```

5、配置nginx相关目录：

```
mkdir -p /fastdfs/cache/nginx/proxy_cache
mkdir -p /fastdfs/cache/nginx/proxy_cache/tmp
```

6、启动nginx。

### 注意事项

1、FastDFS的数据目录和日志目录都在我们配置的 storage.conf和tracker.conf文件中的base_path。
2、作为一个分布式存储，可以设置多个group.
3、tracker的配置基本相同，只有设计到自身IP 的配置才会有差别。
4、Storage的配置只是在指定不同group的地方有区别。