# FastDFS

## FastDFS概述

FastDFS 是一个由 C 语言实现的开源轻量级分布式文件系统，作者余庆(happyfish100)，支持 Linux、FreeBSD、AID 等 Unix 系统，解决了大数据存储和读写负载均衡等问题，适合存储 4KB~500MB 之间的小文件，如图片网站、短视频网站、文档、app 下载站等，UC、京东、支付宝、迅雷、酷狗等都有使用。

FastDFS主要对文件进行存储、同步、上传、下载，有自己的容灾备份（遭遇灾害的时候，系统仍能正常的运行）、负载均衡（把同一个工作任务，分担到不同的人身上，共同完成）、线性扩容（存储容量不够的时候，可以直接增加服务器的数量）机制；

### FastFDS组成

FastDFS架构主要包含Tracker server和Storage server，客户端请求Tracker server进行文件上传、下载的时候，通过Tracker server调度最终由Storage server完成文件上传和下载。

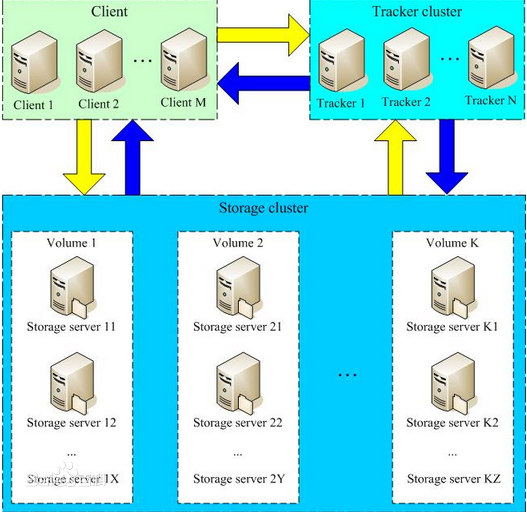
#### Tracker server

Tracker server：跟踪器或者调度器，主要起负载均衡和调度作用。通过Tracker server在文件上传时可以根据一些策略找到Storage server提供文件上传服务。跟踪器中的所有服务器都是对等的，可以根据服务器的压力随时增加或者减少。

#### Storage server

Storage server：存储服务器，作用主要是文件存储，完成文件管理的所有功能。客户端上传的文件主要保存在Storage server上，Storage server没有实现自己的文件系统而是利用操作系统的文件系统去管理文件。

存储服务器采用了分组/分卷的组织方式，整个系统由一个组或者多个组组成；组与组之间的文件是相互独立的；所有组的文件容量累加就是整个存储系统的文件容量；一个组可以由多台存储服务器组成，一个组下的存储服务器中的文件都是相同的，组中的多台存储服务器起到了冗余备份和负载均衡的作用；在组内增加服务器时，如果需要同步数据，则由系统本身完成，同步完成之后，系统自动将新增的服务器切换到线上提供使用；当存储空间不足或者耗尽时，可以动态的添加组。只需要增加一台服务器，并为他们配置一个新的组，即扩大了存储系统的容量。



## FastDFS上传下载流程

### 文件上传流程

客户端上传文件后生成一个file\_id，返回给客户端，客户端利用这个file\_id结合ip地址，生成一个完成图片的url，保存在数据库中。生成的那个file\_id用于以后访问该文件的索引信息。

CLient

Tracker Server

Storage Server

1.定时向tracker上传状态信息

2.上传连接请求

4.返回信息(storage的IP和端口)

3.查询可用storage

5.上传文件(file content和metadatea)

6.生成file\_id

7.将上传内容写入磁盘

8.返回file\_id(路径信息和文件名)

9.存储文件信息

### 文件下载流程

CLient

Tracker Server

Storage Server

1.定时向tracker上传状态信息

2.下载连接请求

4.返回信息(storage的IP和端口)

3.查询可用storage(检验同步状态)

5.file\_id(组名、路径和文件名)

6.查找文件

7.返回file\_content

## 安装FastDFS

### 安装准备

1、上传如下四个文件至Linux虚拟机的/opt目录下

FastDFS\_v5.05.tar.gz

fastdfs-nginx-module\_v1.16.tar.gz

libfastcommonV1.0.7.tar.gz

nginx-1.12.2.tar.gz

2、安装libfastcommon时依赖于libevent、perl和gccl类库，执行如下三个命令安装这三个类库

|  |
| --- |
| [root@xmm opt]# yum install gcc-c++ -y  [root@xmm opt]# yum -y install libevent  [root@xmm opt]# yum install perl\* |

3、解压libfastcommonV1.0.7.tar.gz

|  |
| --- |
| [root@xmm opt]# mkdir libfastcommon  [root@xmm opt]# tar -zxvf libfastcommonV1.0.7.tar.gz -C libfastcommon |

4、编译和安装libfastcommon

|  |
| --- |
| [root@xmm opt]# cd libfastcommon/  [root@xmm libfastcommon]# cd libfastcommon-1.0.7/  [root@xmm libfastcommon-1.0.7]# ./make.sh  [root@xmm libfastcommon-1.0.7]# ./make.sh install |

5、将/usr/lib64目录下的libfastcommon.so文件复制到/usr/lib目录下

|  |
| --- |
| [root@xmm libfastcommon-1.0.7]# cp /usr/lib64/libfastcommon.so /usr/lib |

### 安装FastDFS

1、解压FastDFS\_v5.05.tar.gz压缩包然后进行编译和安装

|  |
| --- |
| [root@xmm opt]# tar -zxvf FastDFS\_v5.05.tar.gz  [root@xmm FastDFS]# ./make.sh  [root@xmm FastDFS]# ./make.sh install |

2、将所有的配置文件复制到/etc/fdfs目录下

|  |
| --- |
| [root@xmm FastDFS]# cd conf  [root@xmm conf]# cp \* /etc/fdfs |

3、新建/opt/fastdfs目录，然后通过vim修改tracker.conf配置文件

|  |
| --- |
| [root@xmm conf]# mkdir /opt/fastdfs  [root@xmm etc]# cd fdfs  [root@xmm fdfs]# vim tracker.conf  ……  # the base path to store data and log files  base\_path=/opt/fastdfs  …… |

4、在/usr/local目录下创建新目录fdfs

|  |
| --- |
| [root@xmm ~]# cd /usr/local/  [root@xmm local]# mkdir fdfs |

5、将解压目录/opt/FastDFS下的stop.sh和restart.sh文件复制到/usr/local/fdfs目录下

|  |
| --- |
| [root@xmm FastDFS]# cp stop.sh /usr/local/fdfs/  [root@xmm FastDFS]# cp restart.sh /usr/local/fdfs/ |

6、通过vim修改目录/etc/init.d下的fdfs\_tracker启动脚本

|  |
| --- |
| [root@xmm init.d]# vim fdfs\_trackerd  ……  PRG=/usr/bin/fdfs\_trackerd  CONF=/etc/fdfs/tracker.conf  ……  if [ ! -f /usr/local/fdfs/stop.sh ]; then  echo "file /usr/local/fdfs/stop.sh does not exist!"  exit 2  fi  if [ ! -f /usr/local/fdfs/restart.sh ]; then  echo "file /usr/local/fdfs/restart.sh does not exist!"  exit 2  fi  ……  start() {  echo -n $"Starting FastDFS tracker server: "  $CMD &  RETVAL=$?  echo  return $RETVAL  }  stop() {  /usr/local/fdfs/stop.sh $CMD  RETVAL=$?  return $RETVAL  }  rhstatus() {  status fdfs\_trackerd  }  restart() {  /usr/local/fdfs/restart.sh $CMD &  } |

7、修改/etc/init.d目录下的fdfs\_trackerd服务的权限并将其注册

|  |
| --- |
| [root@xmm init.d]# chmod 777 fdfs\_trackerd  [root@xmm init.d]# chkconfig --add fdfs\_trackerd  [root@xmm init.d]# chkconfig fdfs\_trackerd on |

8、启动fdfs\_trackerd服务

|  |
| --- |
| [root@xmm init.d]# service fdfs\_trackerd start  Starting fdfs\_trackerd (via systemctl): [ 确定 ] |

### 安装storage

1、通过vim配置/etc/fdfs目录下的storage.conf配置文件

|  |
| --- |
| # the base path to store data and log files  base\_path=/opt/fastdfs  ……  # store\_path#, based 0, if store\_path0 not exists, it's value is base\_path  # the paths must be exist  store\_path0=/opt/fastdfs/fdfs\_storage  #store\_path1=/home/yuqing/fastdfs2  ……  # tracker\_server can ocur more than once, and tracker\_server format is  # "host:port", host can be hostname or ip address  tracker\_server=192.168.235.23:22122  …… |

2、通过vim修改目录/etc/init.d下的fdfs\_storaged启动脚本

|  |
| --- |
| [root@xmm init.d]# vim fdfs\_storaged  ……  PRG=/usr/bin/fdfs\_storaged  CONF=/etc/fdfs/storage.conf  ……  if [ ! -f /usr/local/fdfs/stop.sh ]; then  echo "file /usr/local/fdfs/stop.sh does not exist!"  exit 2  fi  if [ ! -f /usr/local/fdfs/restart.sh ]; then  echo "file /usr/local/fdfs/restart.sh does not exist!"  exit 2  fi  ……  stop() {  /usr/local/fdfs/stop.sh $CMD  RETVAL=$?  return $RETVAL  }  rhstatus() {  status fdfs\_storaged  }  restart() {  /usr/local/fdfs/restart.sh $CMD &  } |

3、修改/etc/init.d目录下的fdfs\_storaged服务的权限并将其注册

|  |
| --- |
| [root@xmm init.d]# chmod 777 fdfs\_trackerd  [root@xmm init.d]# chkconfig --add fdfs\_storaged  [root@xmm init.d]# chkconfig fdfs\_storaged on |

4、启动fdfs\_storaged服务

|  |
| --- |
| [root@xmm init.d]# service fdfs\_storaged start  Starting fdfs\_storaged (via systemctl): [ 确定 ] |

### 上传测试

1、通过vim修改/etc/fdfs目录下的client.conf配置文件

|  |
| --- |
| # the base path to store log files  base\_path=/opt/fastdfs  # tracker\_server can ocur more than once, and tracker\_server format is  # "host:port", host can be hostname or ip address  tracker\_server=192.168.235.23:22122 |

2、测试将/home目录下的jpg文件上传

tracker\_query\_storage\_store\_list\_without\_group: server 1. group\_name=, ip\_addr=192.168.235.23, port=23000表示上传文件的storage服务器和端口号

remote\_filename=M00/00/00/wKjrF1060B2AYVujAABGE98TKuY360.jpg表示上传文件后生成的文件名

example file url: http://192.168.235.23/group1/M00/00/00/wKjrF1060B2AYVujAABGE98TKuY360.jpg

example file url: http://192.168.235 ……1060B2AYVujAABGE98TKuY360\_big.jpg这两个URL都表示web访问文件的URL地址

|  |
| --- |
| [root@xmm FastDFS]# /usr/bin/fdfs\_test /etc/fdfs/client.conf upload /home/dog.jpg  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.  [2019-07-26 18:04:13] DEBUG - base\_path=/opt/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=192.168.235.23, port=23000  group\_name=group1, ip\_addr=192.168.235.23, port=23000  storage\_upload\_by\_filename  group\_name=group1, remote\_filename=M00/00/00/wKjrF1060B2AYVujAABGE98TKuY360.jpg  source ip address: 192.168.235.23  file timestamp=2019-07-26 18:04:13  file size=17939  file crc32=3742575334  example file url: http://192.168.235.23/group1/M00/00/00/wKjrF1060B2AYVujAABGE98TKuY360.jpg  storage\_upload\_slave\_by\_filename  group\_name=group1, remote\_filename=M00/00/00/wKjrF1060B2AYVujAABGE98TKuY360\_big.jpg  source ip address: 192.168.235.23  file timestamp=2019-07-26 18:04:14  file size=17939  file crc32=3742575334  example file url: http://192.168.235.23/group1/M00/00/00/wKjrF1060B2AYVujAABGE98TKuY360\_big.jpg |

## 安装Nginx并结合FastDFS

### 安装fastdfs-nginx-module插件

1、将/opt目录下的fastdfs-nginx-module\_v1.16.tar.gz压缩包解压

|  |
| --- |
| [root@xmm opt]# tar -zxvf fastdfs-nginx-module\_v1.16.tar.gz  fastdfs-nginx-module/  fastdfs-nginx-module/src/  fastdfs-nginx-module/src/ngx\_http\_fastdfs\_module.c  fastdfs-nginx-module/src/mod\_fastdfs.conf  fastdfs-nginx-module/src/config  fastdfs-nginx-module/src/common.h  fastdfs-nginx-module/src/common.c  fastdfs-nginx-module/INSTALL  fastdfs-nginx-module/HISTORY |

2、将/usr/lib64目录下的libfdfsclient.so文件复制到/usr/lib目录下

|  |
| --- |
| [root@xmm opt]# cp /usr/lib64/libfdfsclient.so /usr/lib |

3、修改fastdfs-nginx-module解压目录下的src目录下的config配置文件

|  |
| --- |
| [root@xmm src]# ls  common.c common.h config mod\_fastdfs.conf ngx\_http\_fastdfs\_module.c  [root@xmm src]# vim config  CORE\_INCS="$CORE\_INCS /usr/include/fastdfs /usr/include/fastcommon/"  CORE\_LIBS="$CORE\_LIBS -L/usr/lib -lfastcommon -lfdfsclient" |

4、将fastdfs-nginx-module解压目录下的src目录下的mod\_fastdfs.conf配置文件复制到/etc/fdfsm剥下并修改

|  |
| --- |
| [root@xmm src]# cp mod\_fastdfs.conf /etc/fdfs/  [root@xmm src]# vim /etc/fdfs/mod\_fastdfs.conf  ……  # the base path to store log files  base\_path=/opt/fastdfs  ……  # FastDFS tracker\_server can ocur more than once, and tracker\_server format is  # "host:port", host can be hostname or ip address  # valid only when load\_fdfs\_parameters\_from\_tracker is true  tracker\_server=192.168.235.23:22122  ……  # default value is false  url\_have\_group\_name = true  ……  # store\_path#, based 0, if store\_path0 not exists, it's value is base\_path  # the paths must be exist  # must same as storage.conf  store\_path0=/opt/fastdfs/fdfs\_storage  #store\_path1=/home/yuqing/fastdfs1 |

### 安装Nginx

1、解压Nginx安装包

|  |
| --- |
| [root@xmm opt]# tar -zxvf nginx-1.12.2.tar.gz |

2、安装Nginx，进入Nginx解压目录执行如下语句

pid-path是进程文件，不要放在/var/run目录下

http-client-body-temp-path是HTTP的缓存目录

add-module是nginx加载外接的fdfs插件的模块目录

|  |
| --- |
| [root@xmm opt]# cd nginx-1.12.2/  [root@xmm nginx-1.12.2]# ls  auto CHANGES.ru configure html man src  CHANGES conf contrib LICENSE README  [root@xmm nginx-1.12.2]# ./configure \  > --prefix=/usr/local/nginx \  > --pid-path=/usr/local/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=/opt/fastdfs-nginx-module/src |

3、编译和安装

|  |
| --- |
| [root@xmm nginx-1.12.2]# make  [root@xmm nginx-1.12.2]# make install |

4、配置/usr/local/nginx/conf目录下的nginx.conf配置文件

|  |
| --- |
| [root@xmm conf]# cd /usr/local/nginx/conf  [root@xmm conf]# ls  fastcgi.conf koi-win scgi\_params  fastcgi.conf.default mime.types scgi\_params.default  fastcgi\_params mime.types.default uwsgi\_params  fastcgi\_params.default nginx.conf uwsgi\_params.default  koi-utf nginx.conf.default win-utf  [root@xmm conf]# vim nginx.conf  ……  server {  listen 80;  server\_name 192.168.235.23;  #charset koi8-r;  #access\_log logs/host.access.log main;  location /group1/M00 {  # root html;  # index index.html index.htm;  ngx\_fastdfs\_module;  }  …… |

5、启动/usr/local/nginx/sbin目录下的nginx

|  |
| --- |
| [root@xmm nginx]# ./sbin/nginx  [root@xmm nginx]# ps -ef | grep nginx | grep -v grep  root 4837 1 0 18:59 ? 00:00:00 nginx: master process ./sbin/nginx  nobody 4838 4837 0 18:59 ? 00:00:00 nginx: worker process |

6、配置nginx开机自启，首先在/etc/init.d目录下通过vim创建nginx文件，并将如下内容写入到该文件中，注意nginx和NGINX\_CONF\_FILE的值。

|  |
| --- |
| #!/bin/sh  #  # nginx - this script starts and stops the nginx daemon  #  # chkconfig: - 85 15  # description: NGINX is an HTTP(S) server, HTTP(S) reverse \  # proxy and IMAP/POP3 proxy server  # processname: nginx  # config: /etc/nginx/nginx.conf  # config: /etc/sysconfig/nginx  # pidfile: /var/run/nginx.pid  # Source function library.  . /etc/rc.d/init.d/functions  # Source networking configuration.  . /etc/sysconfig/network  # Check that networking is up.  [ "$NETWORKING" = "no" ] && exit 0  nginx="/usr/local/nginx/sbin/nginx"  prog=$(basename $nginx)  NGINX\_CONF\_FILE="/usr/local/nginx/conf/nginx.conf"  [ -f /etc/sysconfig/nginx ] && . /etc/sysconfig/nginx  lockfile=/var/lock/subsys/nginx  make\_dirs() {  # make required directories  user=`$nginx -V 2>&1 | grep "configure arguments:" | sed 's/[^\*]\*--user=\([^ ]\*\).\*/\1/g' -`  if [ -z "`grep $user /etc/passwd`" ]; then  useradd -M -s /bin/nologin $user  fi  options=`$nginx -V 2>&1 | grep 'configure arguments:'`  for opt in $options; do  if [ `echo $opt | grep '.\*-temp-path'` ]; then  value=`echo $opt | cut -d "=" -f 2`  if [ ! -d "$value" ]; then  # echo "creating" $value  mkdir -p $value && chown -R $user $value  fi  fi  done  }  start() {  [ -x $nginx ] || exit 5  [ -f $NGINX\_CONF\_FILE ] || exit 6  make\_dirs  echo -n $"Starting $prog: "  daemon $nginx -c $NGINX\_CONF\_FILE  retval=$?  echo  [ $retval -eq 0 ] && touch $lockfile  return $retval  }  stop() {  echo -n $"Stopping $prog: "  killproc $prog -QUIT  retval=$?  echo  [ $retval -eq 0 ] && rm -f $lockfile  return $retval  }  restart() {  configtest || return $?  stop  sleep 1  start  }  reload() {  configtest || return $?  echo -n $"Reloading $prog: "  killproc $nginx -HUP  RETVAL=$?  echo  }  force\_reload() {  restart  }  configtest() {  $nginx -t -c $NGINX\_CONF\_FILE  }  rh\_status() {  status $prog  }  rh\_status\_q() {  rh\_status >/dev/null 2>&1  }  case "$1" in  start)  rh\_status\_q && exit 0  $1  ;;  stop)  rh\_status\_q || exit 0  $1  ;;  restart|configtest)  $1  ;;  reload)  rh\_status\_q || exit 7  $1  ;;  force-reload)  force\_reload  ;;  status)  rh\_status  ;;  condrestart|try-restart)  rh\_status\_q || exit 0  ;;  \*)  echo $"Usage: $0 {start|stop|status|restart|condrestart|try-restart|reload|force-reload|configtest}"  exit 2  esac |

然后修改nginx文件的权限并将其注册到服务中

|  |
| --- |
| [root@xmm init.d]# chmod 777 nginx  [root@xmm init.d]# chkconfig --add nginx  [root@xmm init.d]# chkconfig nginx on |

如果Nginx在开机自启时出现找不到pid的错误，那是因为/var/run目录下的Nginx.pid路径被删除，重新设置pid路径即可。

1、修改，/usr/local/nginx/conf/nginx.conf将这一行#pid nginx/logs/nginx.pid的注释去掉，让nginx加载这个目录的pid，不去使用/var/run下的pid。

2、在nginx下创建logs目录，新建一个nginx.pid的文件即可。

## 测试

访问之前上传的图片URL地址，如果可以在浏览器中访问，则说明FastDFS配置成功。