**一、FastDFS简介**

FastDFS是由国人余庆所开发，其项目地址：[https://github.com/happyfish100](https://github.com/happyfish100" \t "_blank)   
 FastDFS是一个轻量级的开源分布式文件系统，主要解决了大容量的文件存储和高并发访问的问题，文件存取时实现了负载均衡。   
 支持存储服务器在线扩容,支持相同的文件只保存一份,节约磁盘。   
 FastDFS只能通过Client API访问，不支持POSIX访问方式。   
 FastDFS适合中大型网站使用，用来存储资源文件(如：图片、文档、视频等)

**二、FastDFS组成部分及其它名词**

**1、tracker server**

跟踪服务器：用来调度来自客户端的请求。且在内存中记录所有存储组和存储服务器的信息状态。

**2、storage server**

存储服务器：用来存储文件(data)和文件属性(metadata)

**3、client**

客户端：业务请求发起方，通过专用接口基于TCP协议与tracker以及storage server进行交互

**group**组，也可称为卷：同组内上的文件是完全相同的

**文件标识**包括两部分：组名和文件名(包含路径)

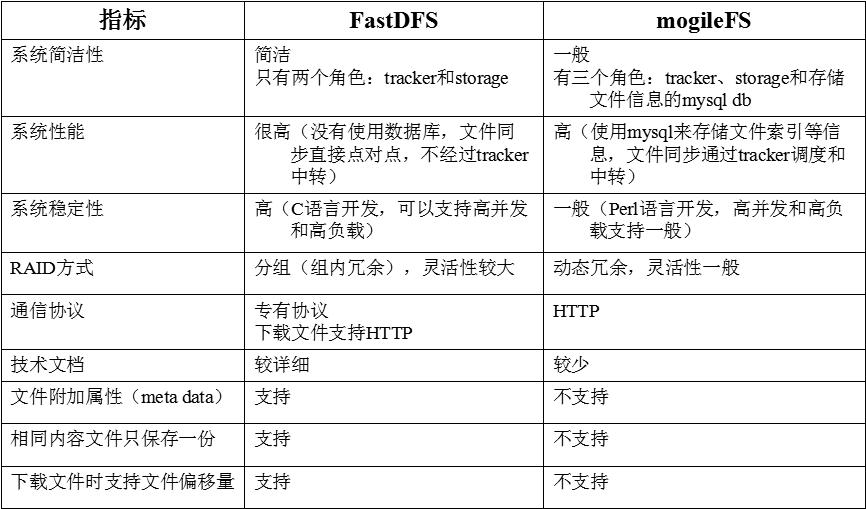
**meta data**文件相关属性：键值对(Key Value Pair)方式

**fid**文件标识符：   
例如：   
group1/M00/00/00/CgEOxVegXB2AdYafAAAB0b8tBbQ9155303   
group\_name：存储组的组名；上传完成后，需要客户端自行保存   
M##：服务器配置的虚拟路径，与磁盘选项store\_path#对应   
两级以两位16进制数字命名的目录   
文件名：与原文件名并不相同；由storage server根据特定信息生成。文件名包含：源存储服务器的IP地址、文件创建时间戳、文件大小、随机数和文件扩展名等

**三、FastDFS同步机制**

1、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；   
 2、文件同步只在同组内的storage server之间进行，采用push方式，即源服务器同步给目标服务器；   
 3、源头数据才需要同步，备份数据不需要再次同步，否则就构成环路了；   
 上述第二条规则有个例外，就是新增加一台storage server时，由已有的一台storage server将已有的所有数据（包括源头数据和备份数据）同步给该新增服务器。

**四、FastDFS与集中存储方式对比**

[](http://obbogqhb1.bkt.clouddn.com/fastdfs2.png)

**五、FastDFS和mogileFS对比**

[](http://obbogqhb1.bkt.clouddn.com/fastdfs1.png)

**六、FastDFS安装配置**

**1、安装开发环境**

# yum -y groupinstall "Development Tools" "Server platform Development"

**2、安装libfastcommon**

# cd /usr/local/

# git clone https://github.com/happyfish100/libfastcommon.git  
# cd libfastcommon/  
# ./make.sh   
# ./make.sh install

**3、安装fastdfs**

# cd /usr/local  
# git clone https://github.com/happyfish100/fastdfs.git  
# cd fastdfs/  
# ./make.sh   
# ./make.sh install

**tracker 配置**

配置文件修改：根据需求修改

# cd /etc/fdfs  
# cp tracker.conf.sample tracker.conf  
# sed -i 's|base\_path=.\*|base\_path=/data/fdfs/tracker|' /etc/fdfs/tracker.conf

启动服务（Centos6）：

# mkdir -pv /data/fdfs/tracker  
# service fdfs\_trackerd start

启动服务（Centos7）方式一：

# mkdir -pv /data/fdfs/tracker  
# /etc/init.d/fdfs\_trackerd start

启动服务（Centos7）方式二：   
添加systemd的units文件

# cat > /usr/lib/systemd/system/fdfs\_trackerd << EOF  
# Systemd unit file for default tomcat  
#   
  
[Unit]  
Description=FastDFS tracker script  
After=syslog.target network.target  
  
[Service]  
Type=notify  
ExecStart=/usr/bin/fdfs\_trackerd /etc/fdfs/tracker.conf

ExecStop=/etc/init.d/fdfs\_trackerd stop

ExecRestart=/etc/init.d/fdfs\_trackerd restart  
  
[Install]  
WantedBy=multi-user.target  
EOF

通过systemd启动

# systemctl start fdfs\_trackerd.service && systemctl enable fdfs\_trackerd.service  
# ss -tnl|grep 22122  
LISTEN     0      128          \*:22122                    \*:\*

**storage 配置**

根据需求修改

# cd /etc/fdfs  
# cp storage.conf.sample storage.conf  
# vim /etc/fdfs/storage.conf  
  group\_name=group1  #指定组名  
  base\_path=/data/fdfs/storage # 用于存储数据  
  store\_path\_count=2 # 设置设备数量  
  store\_path0=/data/fdfs/storage/m0 #指定存储路径0  
  store\_path1=/data/fdfs/storage/m1 #指定存储路径1  
  # 注意：同一组内存储路径不能冲突，例如：下一个节点的存储路径就是m2,m3....等  
  tracker\_server=10.1.14.197:22122 #指定tracker

启动服务（Centos6）：

# mkdir -pv /data/fdfs/storage/{m0,m1} # 创建数据目录  
# service fdfs\_storaged start

启动服务（Centos7）方式一：

# mkdir -pv /data/fdfs/storage/{m0,m1} # 创建数据目录  
# /etc/init.d/fdfs\_storaged start

启动服务（Centos7）方式二：   
添加systemd的units文件

# mkdir -pv /data/fdfs/storage/{m0,m1} # 创建数据目录  
  
# cat > /usr/lib/systemd/system/fdfs\_trackerd << EOF  
# Systemd unit file for default fastdfs storage  
#   
  
[Unit]  
Description=FastDFS storage script  
After=syslog.target network.target  
  
[Service]  
Type=notify  
ExecStart=/usr/bin/fdfs\_storaged /etc/fdfs/storage.conf

ExecStop=/etc/init.d/fdfs\_storaged stop

ExecRestart=/etc/init.d/fdfs\_storaged restart  
  
[Install]  
WantedBy=multi-user.target  
EOF

通过systemd启动

# systemctl start fdfs\_storaged.service && systemctl enable fdfs\_storaged.service

# ss -tnl|grep 23000  
LISTEN     0      128          \*:23000                    \*:\*

**client配置**

修改客户端配置文件

mkdir -pv /data/fdfs/client

base\_path=/data/fdfs/client  
tracker\_server=10.1.14.197:22122

**七、测试FastDFS**

**1、上传文件**

fdfs\_upload\_file   [storage\_ip:port] [store\_path\_index]

# fdfs\_upload\_file /etc/fdfs/client.conf /etc/issue  
group1/M00/00/00/CgEOxVeMxeuABKiEAAAAF30Ccq85795930

**2、查看文件**

fdfs\_file\_info

# fdfs\_file\_info /etc/fdfs/client.conf group1/M00/00/00/CgEOxVeMxeuABKiEAAAAF30Ccq85795930  
  
source storage id: 0  
source ip address: 10.1.14.198  
file create timestamp: 2016-07-18 20:04:59  
file size: 23  
file crc32: 2097312431 (0x7D0272AF)

**3、下载文件**

fdfs\_download\_file   [local\_filename] [ ]

# fdfs\_download\_file /etc/fdfs/client.conf group1/M00/00/00/CgEOxVeMxeuABKiEAAAAF30Ccq85795930 /tmp/issue

**4、查看存储节点状态**

# fdfs\_monitor /etc/fdfs/client.conf  
[2016-08-03 11:58:20] DEBUG - base\_path=/data/fdfs/client, 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  
  
server\_count=1, server\_index=0  
  
tracker server is 10.1.14.197:22122  
  
group count: 1  
  
Group 1:  
group name = mage1  
disk total space = 18898 MB  
disk free space = 17222 MB  
trunk free space = 0 MB  
storage server count = 2  
active server count = 2  
storage server port = 23000  
storage HTTP port = 8888  
store path count = 1  
subdir count per path = 256  
current write server index = 1  
current trunk file id = 0  
  
    Storage 1:  
        id = 10.1.14.198  
        ip\_addr = 10.1.14.198  ACTIVE  
        http domain =   
        version = 5.08  
        join time = 2016-08-03 20:40:46  
        up time = 2016-08-03 11:58:16  
        total storage = 18898 MB  
        free storage = 17222 MB  
        upload priority = 10  
        store\_path\_count = 1  
        subdir\_count\_per\_path = 256  
        storage\_port = 23000  
        storage\_http\_port = 8888  
        current\_write\_path = 0  
        source storage id =   
        if\_trunk\_server = 0  
        connection.alloc\_count = 256  
        connection.current\_count = 0  
        connection.max\_count = 0  
        total\_upload\_count = 1  
        success\_upload\_count = 1  
        total\_append\_count = 0  
        success\_append\_count = 0  
        total\_modify\_count = 0  
        success\_modify\_count = 0  
        total\_truncate\_count = 0  
        success\_truncate\_count = 0  
        total\_set\_meta\_count = 0  
        success\_set\_meta\_count = 0  
        total\_delete\_count = 0  
        success\_delete\_count = 0  
        total\_download\_count = 0  
        success\_download\_count = 0  
        total\_get\_meta\_count = 0  
        success\_get\_meta\_count = 0  
        total\_create\_link\_count = 0  
        success\_create\_link\_count = 0  
        total\_delete\_link\_count = 0  
        success\_delete\_link\_count = 0  
        total\_upload\_bytes = 23  
        success\_upload\_bytes = 23  
        total\_append\_bytes = 0  
        success\_append\_bytes = 0  
        total\_modify\_bytes = 0  
        success\_modify\_bytes = 0  
        stotal\_download\_bytes = 0  
        success\_download\_bytes = 0  
        total\_sync\_in\_bytes = 0  
        success\_sync\_in\_bytes = 0  
        total\_sync\_out\_bytes = 0  
        success\_sync\_out\_bytes = 0  
        total\_file\_open\_count = 1  
        success\_file\_open\_count = 1  
        total\_file\_read\_count = 0  
        success\_file\_read\_count = 0  
        total\_file\_write\_count = 1  
        success\_file\_write\_count = 1  
        last\_heart\_beat\_time = 2016-08-03 11:58:15  
        last\_source\_update = 2016-08-03 20:43:55  
        last\_sync\_update = 1970-01-01 08:00:00  
        last\_synced\_timestamp = 1970-01-01 08:00:00   
    Storage 2:  
        id = 10.1.14.199  
        ip\_addr = 10.1.14.199  ACTIVE  
        http domain =   
        version = 5.08  
        join time = 2016-08-03 20:42:13  
        up time = 2016-08-03 11:58:08  
        total storage = 18898 MB  
        free storage = 17222 MB  
        upload priority = 10  
        store\_path\_count = 1  
        subdir\_count\_per\_path = 256  
        storage\_port = 23000  
        storage\_http\_port = 8888  
        current\_write\_path = 0  
        source storage id = 10.1.14.198  
        if\_trunk\_server = 0  
        connection.alloc\_count = 256  
        connection.current\_count = 0  
        connection.max\_count = 0  
        total\_upload\_count = 0  
        success\_upload\_count = 0  
        total\_append\_count = 0  
        success\_append\_count = 0  
        total\_modify\_count = 0  
        success\_modify\_count = 0  
        total\_truncate\_count = 0  
        success\_truncate\_count = 0  
        total\_set\_meta\_count = 0  
        success\_set\_meta\_count = 0  
        total\_delete\_count = 0  
        success\_delete\_count = 0  
        total\_download\_count = 0  
        success\_download\_count = 0  
        total\_get\_meta\_count = 0  
        success\_get\_meta\_count = 0  
        total\_create\_link\_count = 0  
        success\_create\_link\_count = 0  
        total\_delete\_link\_count = 0  
        success\_delete\_link\_count = 0  
        total\_upload\_bytes = 0  
        success\_upload\_bytes = 0  
        total\_append\_bytes = 0  
        success\_append\_bytes = 0  
        total\_modify\_bytes = 0  
        success\_modify\_bytes = 0  
        stotal\_download\_bytes = 0  
        success\_download\_bytes = 0  
        total\_sync\_in\_bytes = 23  
        success\_sync\_in\_bytes = 23  
        total\_sync\_out\_bytes = 0  
        success\_sync\_out\_bytes = 0  
        total\_file\_open\_count = 1  
        success\_file\_open\_count = 1  
        total\_file\_read\_count = 0  
        success\_file\_read\_count = 0  
        total\_file\_write\_count = 1  
        success\_file\_write\_count = 1  
        last\_heart\_beat\_time = 2016-08-03 11:58:07  
        last\_source\_update = 1970-01-01 08:00:00  
        last\_sync\_update = 2016-08-03 20:43:57  
        last\_synced\_timestamp = 2016-08-03 20:43:55 (0s delay)

**八、配置nginx为storage server提供http访问接口**

**1、下载fastdfs-nginx-module**

# cd /usr/local

# git clone https://github.com/happyfish100/fastdfs-nginx-module.git

**2、下载nginx源码，并编译支持fastdfs**

# 安装依赖程序  
# yum -y install openssl-devel pcre-devel   
# wget http://nginx.org/download/nginx-1.12.2.tar.gz  
# tar -zxvf nginx-1.12.2.tar.gz  
  
# cd nginx-1.12.2/  
# useradd -r nginx

# mkdir -pv /var/log/nginx/  
# ./configure --prefix=/usr/local/nginx --conf-path=/etc/nginx/nginx.conf --error-log-path=/var/log/nginx/error.log --http-log-path=/var/log/nginx/access.log --pid-path=/var/run/nginx/nginx.pid --lock-path=/var/lock/nginx.lock --user=nginx --group=nginx --with-http\_ssl\_module --with-http\_stub\_status\_module --with-pcre --add-module=../fastdfs-nginx-module/src  
# make  
# make install

**3、复制配置文件**

# cp /usr/local/fastdfs-nginx-module/src/mod\_fastdfs.conf /etc/fdfs/  
# cp /usr/local/fastdfs/conf/{http.conf,mime.types} /etc/fdfs/

注意：fastdfs-5.0.8为fastdfs源码目录，如为更改，应叫fastdfs

**4、配置fastdfs-nginx-module配置文件**

# vim /etc/fdfs/mod\_fastdfs.conf  
base\_path=/data/fdfs/storage #存储节点的目录位置  
tracker\_server=10.1.14.197:22122 #制定tracker-server  
storage\_server\_port=23000  
group\_name=mage1  #制定组名  
url\_have\_group\_name = true  #访问路径中是否包括组名  
store\_path\_count=1 #配置路径个数  
store\_path0=/data/fdfs/storage/m0  #指定要查看的路径  
  
cat >> /etc/fdfs/mod\_fastdfs.conf <<EOF

[group1]

group\_name=group1

storage\_server\_port=23000

store\_path\_count=1

store\_path0=/data/fdfs/storage/m0

EOF

**5、配置nginx**

# vim /etc/nginx/nginx.conf  
location ~ /group[0-9]+/M00/ {  
    root /data/fdfs/storage/m0/data/;  
    ngx\_fastdfs\_module;  
}         
# cat >> /etc/profile.d/nginx.sh << EOF  
export PATH=$PATH:/usr/local/nginx/sbin  
EOF  
  
# source /etc/profile.d/nginx.sh

**6、为存储文件路径穿件链接至M00**

# ln -sv /data/fdfs/storage/m0/data  /data/fdfs/storage/m0/data/M00

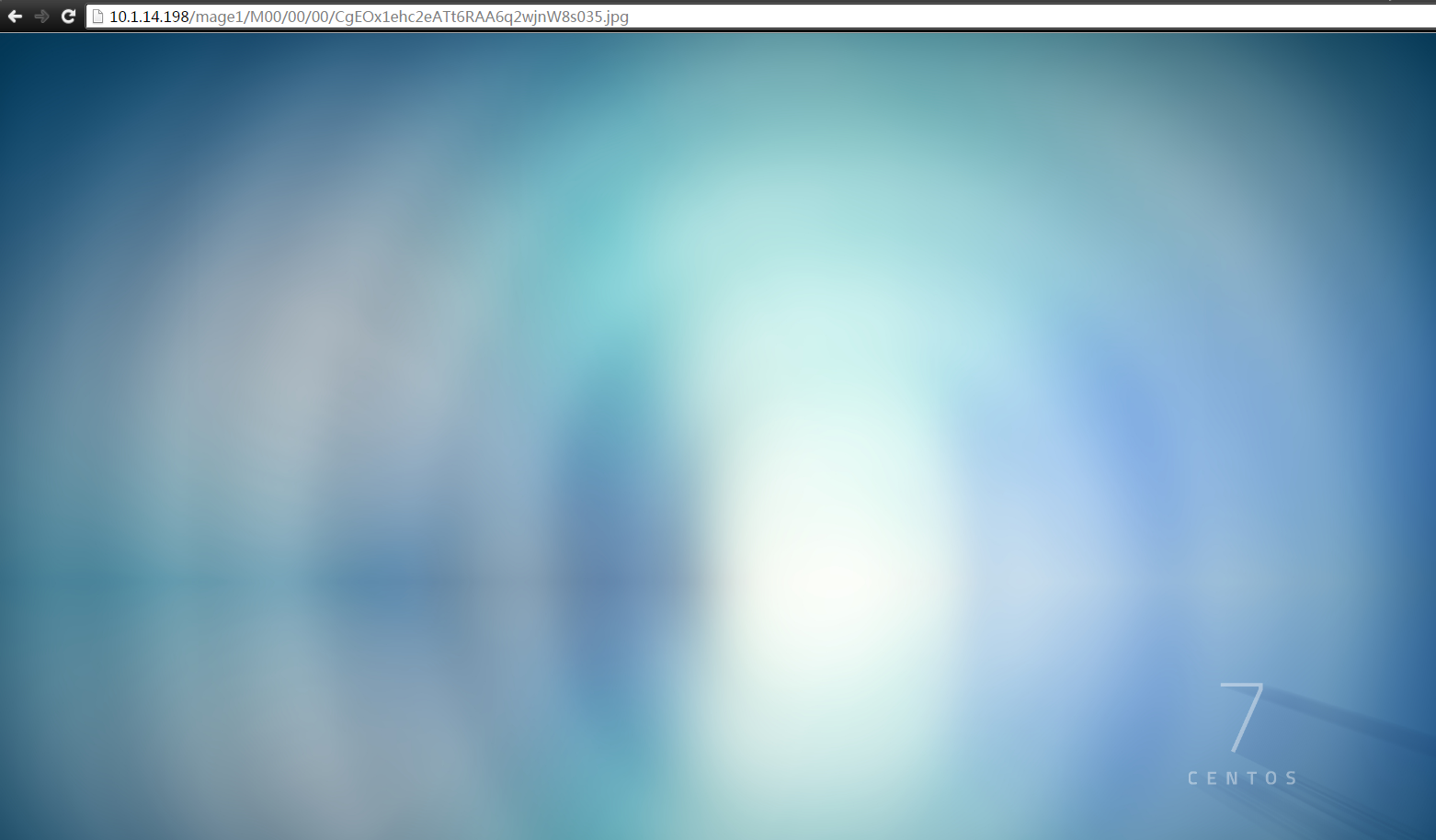
**7、启动nginx和重启storage并上传文件测试**

启动nginx  
# nginx -t  
# nginx  
# /etc/init.d/fdfs\_storaged restart  
# ss -tnl|grep -E "(80|23000)"  
LISTEN     0      128          \*:80                       \*:\*                    
LISTEN     0      128          \*:23000                    \*:\*

**上传文件**

# fdfs\_upload\_file /etc/fdfs/client.conf /usr/share/wallpapers/CentOS7/contents/images/2560x1600.jpg  
group1/M00/00/00/CgEOx1ehc2eATt6RAA6q2wjnW8s035.jpg

**查看测试**

[](http://obbogqhb1.bkt.clouddn.com/fastdfs3.png)