# FASTDFS

# 为什么使用FASTDFS

|  |
| --- |
| Fast DFS(最快的分布式文件系统)  使用FASTDFS的公司有：  支付宝，京东商城，赶集网，淘米网，迅雷，UC等等。  目前规模最大的一家：集群中存储的group数有200个，存储容量达到3PB，文件数接近1亿个，目前处于持续增长中。 |

# 分布式文件系统介绍

|  |
| --- |
| FastDFS是淘宝的余庆先生所开发，是一款开源的轻量级分布式文件系统，采用纯C实现，目前提供了C,Java和PHP的API,主要用它来对文件进行管理，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。同类的分布式文件系统有谷歌的GFS、HDFS（Hadoop）、TFS（淘宝）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。  存储节点主要的工作：存储、同步和提供存取接口。  整体来说：FastDFS主要有以下两个主要特点：   1. 高可靠性：无单点故障； 2. 高吞吐量：只要 Group 足够多，数据流量是足够分散的。   FastDFS系统结构如下图所示：  IMG_256  工作原理说明：  跟踪器和存储节点都可以由一台或多台服务器构成。跟踪器和存储节点中的服务器均可以随时增加或下线而不会影响线上服务。其中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情况随时增加或减少。 为了支持大容量，存储节点（服务器）采用了分卷（或分组）的组织方式。存储系统由一个或多个卷组成，卷与卷之间的文件是相互独立的，所有卷 的文件容量累加就是整个存储系统中的文件容量。一个卷可以由一台或多台存储服务器组成，一个卷下的存储服务器中的文件都是相同的，卷中的多台存储服务器起 到了冗余备份和负载均衡的作用。  在卷中增加服务器时，同步已有的文件由系统自动完成，同步完成后，系统自动将新增服务器切换到线上提供服务。  当存储空间不足或即将耗尽时，可以动态添加卷。只需要增加一台或多台服务器，并将它们配置为一个新的卷，这样就扩大了存储系统的容量。 |

# 上传文件流程

|  |
| --- |
| FastDFS中的文件标识分为两个部分：卷名和文件名，二者缺一不可。  IMG_257                                   FastDFS file upload  上传文件交互过程： 1. client询问tracker可以执行上传操作的storage，不需要附加参数； 2. tracker返回一台可用的storage； 3. client直接和storage通讯完成文件上传。 |

# 下载文件流程

|  |
| --- |
| IMG_258                             FastDFS file download  下载文件交互过程：  1. client询问tracker下载文件的storage，参数为文件标识（卷名和文件名）； 2. tracker返回一台可用的storage； 3. client直接和storage通讯完成文件下载。  需要说明的是，client为使用FastDFS服务的调用方，client也应该是一台服务器，它对tracker和storage的调用均为服务器间的调用 |

# 搭建图片服务器

## 安装依赖的环境

|  |
| --- |
| FastDFS是C语言开发，安装FastDFS需要先从官网下载源码，然后进行编译，但编译依赖gcc环境，如果没有gcc环境，需要安装gcc：  Rpma--->rpmb--->rpmc yum a  yum -y install gcc-c++  Yum  FastDFS依赖libevent库，需要安装：  yum -y install libevent  可能遇到的问题:  Another app is currently holding the yum lock; waiting for it to exit...  **建议：先等待一段时间，让系统自动解锁。如果长时间没有反应，再采用下面的方案**  another app is currently holding the yum lock;waiting for it to exit解决      有时用yum升级一些文件时，会出现以下情况：        another app is currently holding the yum lock;waiting for it to exit...      可以通过强制关掉yum进程：        #rm -f /var/run/yum.pid      然后就可以使用yum了。 |

## 上传安装文件

|  |
| --- |
| 上传到/usr/local目录下 |

## 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。  cd /usr/local  tar -zxvf libfastcommonV1.0.7.tar.gz  cd libfastcommon-1.0.7  ./make.sh 编译  ./make.sh install 安装  **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**    查找当前目录是否包含指定文件  find -name libfastcommon.so  find /usr/lib -name libfastcommon.so |

## 安装FastDFS\_v5.05（tracker+storage）

|  |
| --- |
| 注意，此处我们将这两个东西配置在一台服务器，真实环境考虑高可用，应该是部署在多台服务器    1，解压FastDFS\_v5.05.tar.gz  tar -zxvf FastDFS\_v5.05.tar.gz  2，cd FastDFS  ./make.sh  ./make.sh install  安装成功将安装目录下的conf下的文件拷贝到/etc/fdfs/下。（nginx） |

### 安装Tracker服务

|  |
| --- |
| 配置tracker服务。修改/etc/fdfs /tracker.conf文件。   启动tracker。/usr/bin/fdfs\_trackerd /etc/fdfs/tracker.conf  重启使用命令：/usr/bin/fdfs\_trackerd /etc/fdfs/tracker.conf restart  配置开机自启动  将以上命令加入到该配置文件中  vi /etc/rc.d/rc.local  注意：配置完毕之后要将ip设置为静态IP |

### 安装Storage服务

|  |
| --- |
| 1. 如果是在不同的服务器安装，则前面的两步需要重新执行。 2. 配置storage服务。修改/etc/fdfs /storage.conf文件        1. 启动storage服务，此处会时间长一些   /usr/bin/fdfs\_storaged /etc/fdfs/storage.conf **restart**  配置开机自启动  将以上命令加入到该配置文件中  vi /etc/rc.d/rc.local |

## 配置客户端

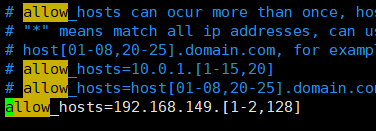
|  |
| --- |
| 1. 将/usr/local/FastDFS/client 里面的libfdfsclient.so 拷贝到 /usr/lib 这一步非常重要（nginx）      1. 修改配置文件/etc/fdfs/client.conf      1. 测试   /usr/bin/fdfs\_test /etc/fdfs/client.conf upload /home/hgz/shuai.jpg |

## 安装nginx及nginx插件

|  |
| --- |
| nginx需要依赖一些环境   * **gcc（前面已经安装，所以不用再安装）**   yum install gcc-c++   * PCRE   **yum install -y pcre pcre-devel**   * zlib   **yum install -y zlib zlib-devel**   * openssl   **yum install -y openssl openssl-devel**  **安装nginx，和nginx插件，两者结合**  可以使用官方提供的nginx插件。要使用nginx插件需要重新编译。  fastdfs-nginx-module\_v1.16.tar.gz   1. 解压插件压缩包 2. 修改fastdfs-nginx-module/src/config文件，把其中的local去掉。      1. 把fastdfs-nginx-module/src/mod\_fastdfs.conf文件复制到/etc/fdfs目录下。并编辑该文件：        1. 解压nginx，进入nginx目录 2. 对nginx重新config   ./configure --add-module=/usr/local/soft/fastdfs-nginx-module/src   1. make 2. make install   **注意：nginx默认安装到了/usr/local/nginx目录下，所以接下来要进入此目录中**   1. nginx的配置 cd /usr/local/nginx/conf   在nginx的配置文件中添加一个location：     1. 启动nginx 2. 进入/usr/local/nginx/sbin目录 3. [root@localhost sbin]# ./nginx     可以再次上传一张图片，然后根据生成的路径进行访问  关闭防火墙  简单的做法（实验的时候就行）  service iptables stop  /usr/bin/fdfs\_test /etc/fdfs/client.conf upload /home/hgz/mei.jpg  http://192.168.10.163/group1/M00/00/00/wKgGhFhNXuuAEWPuAAAwpEw7Ci8140\_big.jpg  http://192.168.10.163/group1/M00/00/00/wKgGhFhNXuuAEWPuAAAwpEw7Ci8140.jpg  正确做法  vim /etc/sysconfig/iptables  service iptables restart  到此，服务器已经可以正常存储文件和提供访问服务，剩下的就是Java客户端要做的事了 |

# 6.给服务器配置访问白名单

vim /etc/fdfs/tracker.conf



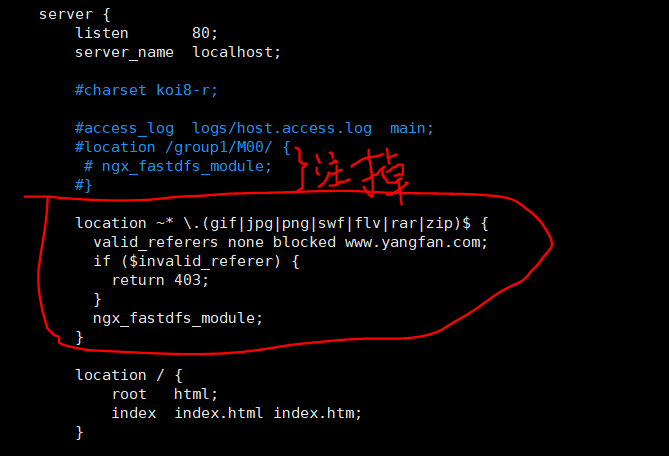
然后重启tracker和storage

# 7.配置Nginx防盗链

作用：别人的网站无法访问我们的图片链接

服务器端修改nginx的配置文件

vim /usr/local/nginx/conf/nginx.conf



location ~\* \.(gif|jpg|png|swf|flv|rar|zip)$ {

valid\_referers none blocked www.yangfan.com;

if(!valid\_referers){

return 403;

}

ngx\_fastdfs\_module;

}

重启nginx服务

./nginx -s reload