# 今日任务

1. 海量文件存储讨论（重要）
2. Fastdfs理论
3. 源码编译、安装、配置fastdfs（重点）
4. 本地客户端文件上传测试

# 图片存储

分析图片文件的特点？？

二进制格式，容量小，很多，零碎。

这样一个文件放在哪里合适呢？？

数据库 ??? 可以，使用Blob可以存储。 存在形式：数据库字段形式。

硬盘？？？可以，存在形式：文件。

哪种方式更适合我们的项目？？ 这里应该直接以操作系统的文件形式进行存储，性能会更高点。

可以将图片访问路径保存在数据库里，因为路径可以很轻松访问，放在数据库也方便管理。

1. 单体架构（单机）

应用服务器

图片保存目录

Upload :

1.jpg

上传、浏览

存储容量有限。访问不会出现问题，并发高时，需要做集群。

2） 单体架构（集群）

应用服务器

图片保存目录

Upload :

2.jpg

上传、浏览

应用服务器

图片保存目录

Upload :

1.jpg

同步？？

负载均衡

有何问题？？这种方式，会出现上传文件和浏览文件不在同一台服务器上的问题，导致明明上传了文件，却访问不到文件。解决办法：考虑做文件同步，或者应用服务器和文件服务器分离。

3） 集中式文件服务器

应用服务器

应用服务器

负载均衡

访问网页

图片服务器

www

images

1.jpg

上传、下载

应用和图片分离，访问图片没有问题，但是**单机容量有限**，其次存在**单点故障**问题，最后就是**并发性**有限。

4）集中式文件服务器（集群）

应用服务器

应用服务器

负载均衡

访问网页

图片服务器

www

images

2.jpg

上传、下载

图片服务器

www

images

1.jpg

负载均衡

**同步**

**？**

该结构能解决海量文件存储的问题，并发性也可水平扩展，但是文件访问问题依然存在，所以文件同步问题依然存在，普通的文件系统已经不满足这种需求了，所以亟待一种**新的文件系统来专门解决**。

# 分布式存储

1. 对象存储（Object Storage Service，简称 OSS），目前由各大云服务商提供，特点：简单，易用，不需要你维护，api接口简单好用，关键是文档齐全，程序员不需要担心使用问题。

缺点：要钱。其次，不利于学习，毕竟什么都给你封装好了。不利于技术的提高。

**推荐大家去企业用这个，哈哈。省事啊！！！**

1. 分布式文件系统（开源）

由于开源，所以服务器什么的，都得自己从头开始搞，而且肯定会遇到坑，这时逼着看文档，要么啃源码，总之，虽然辛苦，但能学习到很多东西，有利于技术的提高。另外，老板喜欢吧，因为不要钱啊，不过你苦啊，出了问题，还得你来啊。

**讲实话，这个逼不好装，但是我们现在毕竟是学习阶段，再难的逼，哭着也要装下去！准备好了吗？Let`s go !!!!**

# Fastdfs

## 简介

阿里，大名鼎鼎的**余庆杰作**。国内创业公司，电商公司使用较多，例如，京东初期的图片服务器就是基于此。

FastDFS是一个**轻量级**的开源分布式文件系统

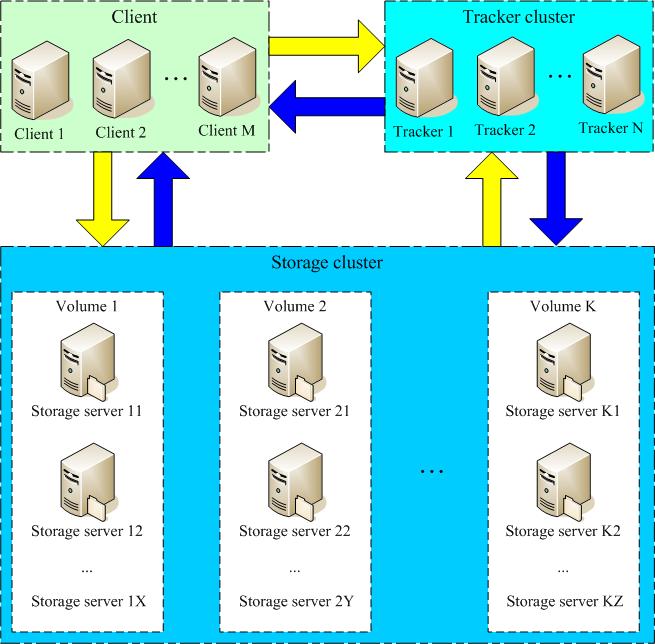
FastDFS主要解决了大容量的文件存储和高并发访问的问题，文件存取时实现了负载均衡

FastDFS实现了软件方式的RAID，可以使用廉价的IDE硬盘进行存储，支持存储服务器在线扩容，支持相同内容的文件只保存一份，节约磁盘空间

FastDFS只能通过Client API访问，不支持POSIX访问方式

FastDFS特别适合**大中型网站**使用，用来存储资源文件（如：图片、文档、音频、视频等等）

## 架构



Tracker Server：跟踪服务器，主要做调度工作，在访问上起负载均衡的作用。记录storage server的状态，是连接Client和Storage server的枢纽。

Storage Server：存储服务器，文件和meta data都保存到存储服务器上

group：组，也可称为卷。同组内服务器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，如：width=1024,heigth=768

## 同步

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；

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；

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。

**知道大家都不喜欢学理论，但是真正有用的其实就是这些理论，把理论弄透了，你完全可以实现一个 java版的（这个是C语言写的），那我们下面就动动手吧！**

# 实操

## 准备虚拟机



本文虚拟机ip地址网段为：**192.168.192.\***

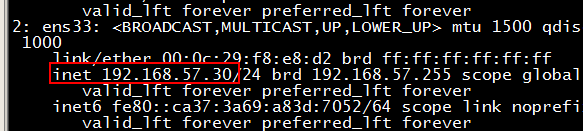
## 补充（修改虚拟机）

**hostnamectl set-hostname storage**

**/etc/sysconfig/network-scripts/ifcfg-ens33**

**systemctl restart network**

**查看网卡修改后结果： ip address**

****

## 安装libfastcommon-v1.0.36

**yum install wget -y**

**wget https://github.com/happyfish100/libfastcommon/archive/V1.0.36.tar.gz**

**yum install gcc perl –y**

**tar zxf V1.0.36.tar.gz**

**cd libfastcommon-1.0.36/**

**./make.sh**

**./make.sh install**

以上tracker和storage服务器安装步骤相同！！

## 安装fastdfs-v5.11

**wget https://github.com/happyfish100/fastdfs/archive/V5.11.tar.gz**

**tar zxf V5.11.tar.gz**

**cd fastdfs-5.11/**

**./make.sh**

**./make.sh install**

以上tracker和storage服务器安装步骤相同！！

## 配置fastdfs-tracker

**创建默认数据和文件保存基路径**：

mkdir -p /fastdfs/tracker

***复制配置文件***：

cd /etc/fdfs

cp tracker.conf.sample tracker.conf

编辑配置文件：

vi /etc/fdfs/tracker.conf

|  |
| --- |
| bind\_addr=192.168.192.20  # the tracker server port  port=22122  # the base path to store data and log files  base\_path=/fastdfs/tracker |

修改防火墙(7.0)：

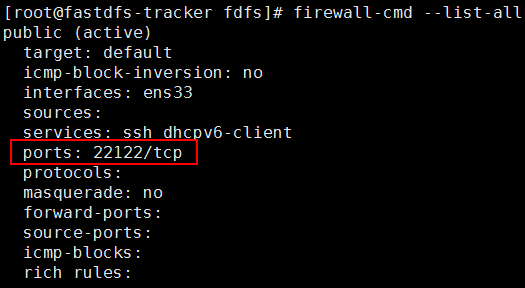
firewall-cmd --zone=public --add-port=22122/tcp --permanent

**重载防火墙规则：**

firewall-cmd --reload

**查看规则：**

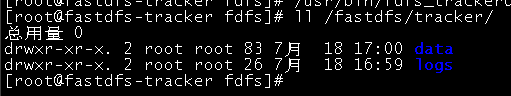
firewall-cmd --list-all



启动：

/usr/bin/fdfs\_trackerd /etc/fdfs/tracker.conf start

启动成功后会在/fastdfs/tracker下创建**data,logs**目录

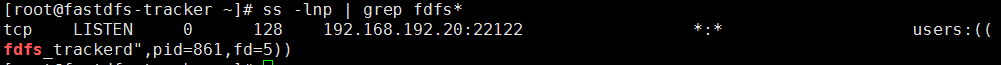


查看：ps aux | grep fdfs\*



端口占用：

ss -lnpt | grep fdfs\*



关闭：/usr/bin/fdfs\_trackerd /etc/fdfs/tracker.conf stop

**开机自启动**：

添加执行权限：chmod +x /etc/rc.d/rc.local

vi /etc/rc.d/rc.local

添加这一行，/usr/bin/fdfs\_trackerd /etc/fdfs/tracker.conf start

测试需要重启电脑，查看进程

## 配置fastdfs-storage

cd /etc/fdfs/

cp storage.conf.sample storage.conf

vi storage.conf

|  |
| --- |
| # the base path to store data and log files  base\_path=/fastdfs/storage  # store\_path#, based 0, if store\_path0 not exists, it's value is base\_path  # the paths must be exist  store\_path0=/fastdfs/storage  # tracker\_server can ocur more than once, and tracker\_server format is  # "host:port", host can be hostname or ip address  tracker\_server=192.168.221.3:22122（这里的ip地址改成自己电脑的tracker）  # the port of the web server on this storage server  http.server\_port=8888 |

mkdir -p /fastdfs/storage

打开防火墙：

firewall-cmd --zone=public --add-port=23000/tcp --permanent

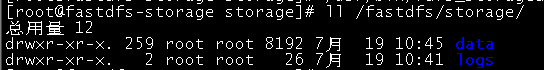
重启防火墙：

firewall-cmd --reload

启动服务(***请确定tracker服务已经启动***)：

/usr/bin/fdfs\_storaged /etc/fdfs/storage.conf start

启动成功后会在/fastdfs/storage 下创建data,logs目录



关闭：/usr/bin/fdfs\_storaged /etc/fdfs/storage.conf stop

开机自启动：vi /etc/rc.d/rc.local

添加这一行，/usr/bin/fdfs\_storaged /etc/fdfs/storage.conf start

添加执行权限：chmod +x /etc/rc.d/rc.local

## 上传文件测试（在tracker上）

cd /etc/fdfs/

cp client.conf.sample client.conf

vi client.conf

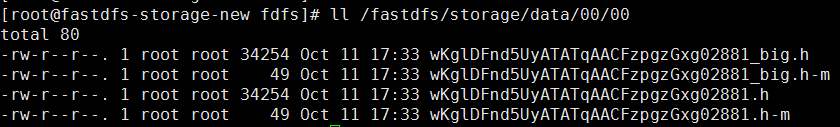
|  |
| --- |
| # the base path to store log files  base\_path=/fastdfs/tracker  # tracker\_server can ocur more than once, and tracker\_server format is  # "host:port", host can be hostname or ip address  tracker\_server=192.168.221.3:22122（修改成自己的电脑的地址） |

保存退出：x

测试程序：

/usr/bin/fdfs\_test /etc/fdfs/client.conf upload /usr/include/stdlib.h

查看storage服务器上是否有文件上传了：



如上图，上传文件成功。

## storage server安装nginx

下载解压nginx-1.12.2

cd /root

wget http://nginx.org/download/nginx-1.12.2.tar.gz

tar zxf nginx-1.12.2.tar.gz

下载解压fastdfs\_nginx\_module



下载fastdfs\_nginx\_module.zip



当前最新的master不能下了与fastdfs不兼容，最新的发布版也不能下载，也不兼容，所以只能寻找以前的提交版本：至少是这个提交之前的，



wget https://github.com/happyfish100/fastdfs-nginx-module/archive/85347be190dcfe9d46bd67176f84afaa32718142.zip

yum install unzip -y

unzip 85347be190dcfe9d46bd67176f84afaa32718142.zip

安装nginx

yum install -y pcre pcre-devel zlib zlib-devel openssl openssl-devel

cd /root/nginx-1.12.2

./configure --add-module=/root/fastdfs-nginx-module-85347be190dcfe9d46bd67176f84afaa32718142/src/

make

make install

## 配置nginx

vi /usr/local/nginx/conf/nginx.conf

location /M00 {

root /fastdfs/storage/data;

ngx\_fastdfs\_module;

}

ln -s /fastdfs/storage/data /fastdfs/storage/data/M00

cd /root/fastdfs-nginx-module-85347be190dcfe9d46bd67176f84afaa32718142/src/

cp mod\_fastdfs.conf /etc/fdfs/

cd /etc/fdfs/

vi mod\_fastdfs.conf

|  |
| --- |
| connect\_timeout=10  base\_path=/tmp（默认为/tmp）  tracker\_server=172.18.10.232:22122  storage\_server\_port=23000（默认配置为23000）  url\_have\_group\_name = false  store\_path0=/fastdfs/storage  group\_name=group1（默认配置为group1） |

cd /root/fastdfs-5.11/conf

cp http.conf mime.types /etc/fdfs/

开放http访问服务：

firewall-cmd --zone=public --add-service=http --permanent

firewall-cmd --reload

启动nginx

/usr/local/nginx/sbin/nginx

浏览器输入：http://192.168.192.30/M00/00/00/wKjAHlrsFgWATavhAACE7ieMB5429600.h

下载成功！！

开机自启动：vi /etc/rc.d/rc.local

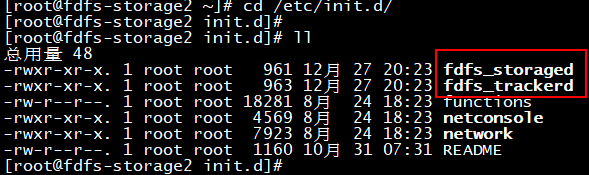
添加这一行，/usr/local/nginx/sbin/nginx

(-----------------------------------------------------the end----------------------------------------)

**根据以往经验，能够一次性成功的同学，几乎为零！！！！所以同学们，一次没成功的，那就again吧。学习使我快乐，C:\Users\ADMINI~1\AppData\Local\Temp\SGPicFaceTpBq\11800\FDD0985E.png C:\Users\ADMINI~1\AppData\Local\Temp\SGPicFaceTpBq\11800\FDD0AEAB.png。**

# 备注

关于启动fdfs的方式，其实还有一种更简单的方式，就是利用服务来启动，有点类似于windows的服务管理系统。Systemctl是7.x后提供的服务管理功能，不过该功能依赖于启动脚本，好在fdfs已经写好了，而且默认也安装好了，只需要开启就行了。如图所示

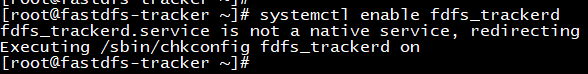


启动服务：**systemctl start fdfs\_storaged**

停止服务：**systemctl stop fdfs\_storaged**

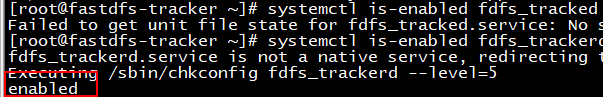
重新启动：**systemctl restart fdfs\_storaged**

开机自启：**systemctl enable fdfs\_storaged**



检查单元是否配置为自己主动启动：

$ systemctl is-enabled <单元>



开机自己主动激活单元：

# systemctl enable <单元>

取消开机自己主动激活单元：

# systemctl disable <单元>

**其它命令**

**# ./nginx -s stop**

**# ./nginx -s quit**

**# ./nginx -s reload**

重启nginx：-s reload

删除端口：firewall-cmd --zone=public --remove-port=12345/tcp –permanent

远程拷贝文件：从本机拷贝到远程服务器，反过来参数互换位置即可。

**scp nginx-1.15.7.tar.gz 85347be190dcfe9d46bd67176f84afaa32718142.zip** [**root@192.168.19.128:/root**](mailto:root@192.168.19.128:/root)

源码地址：<https://github.com/happyfish100>

# 5.12

如果一定想用最新版，一定要先仔细看官网说明，版本之间的兼容性！！

官网下载当前的master就是5.12版本，目前还没发布出来，截至本文写作时（2018-12-26）。

wget https://codeload.github.com/happyfish100/fastdfs/zip/master

yum install unzip -y

unzip master

mv fastdfs-master fastdfs-5.12