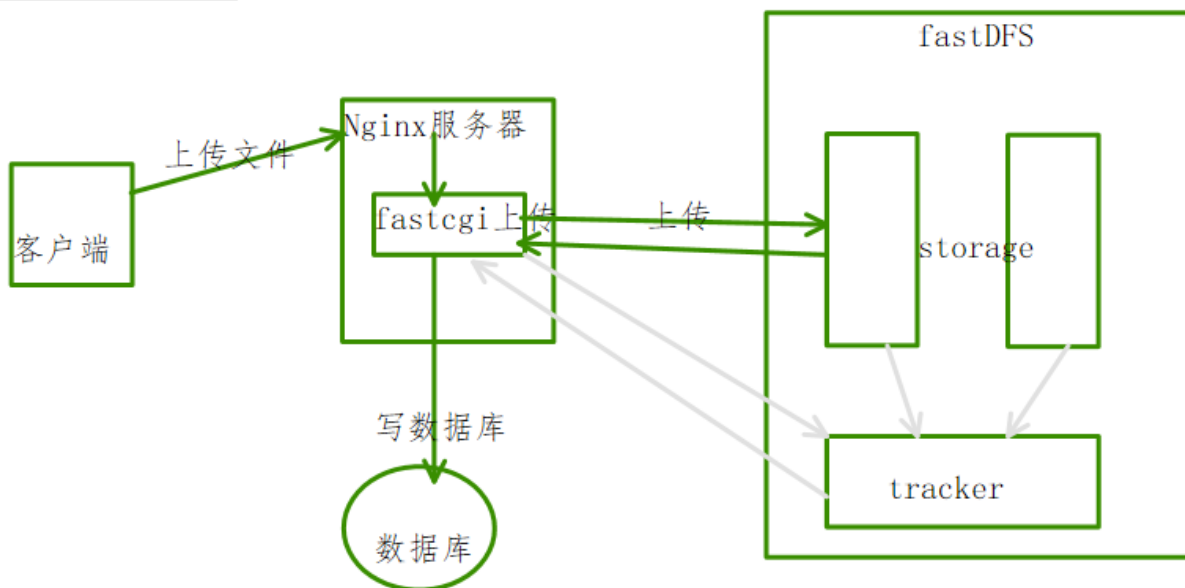
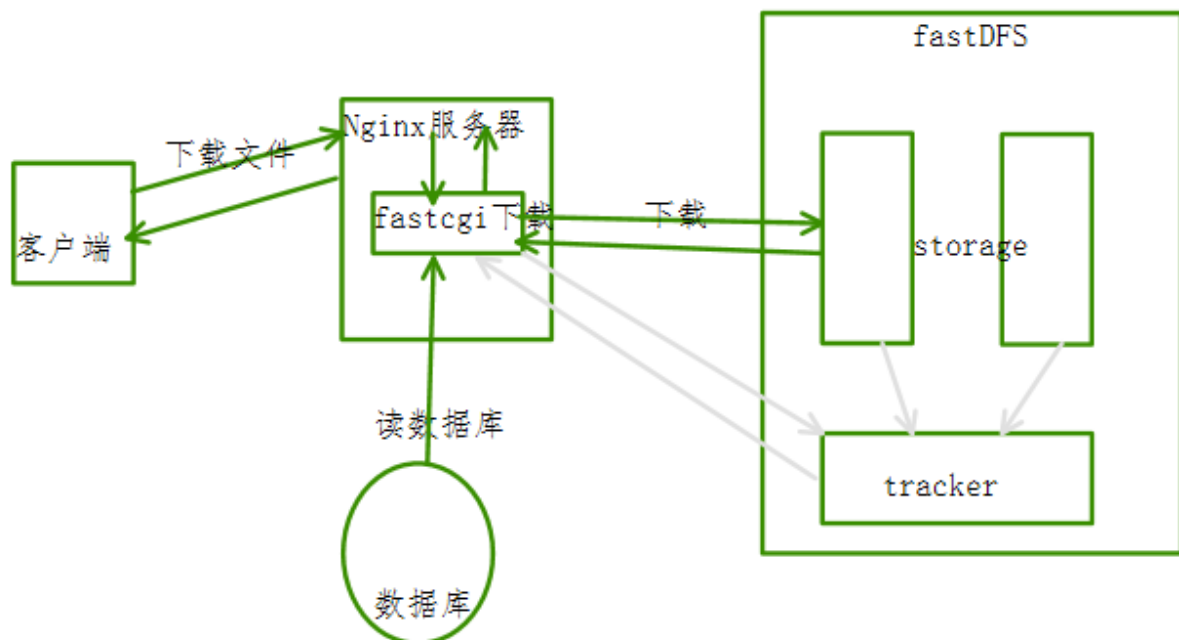


1. 文件上传下载流程

1. 文件上传流程



2. 文件下载流程



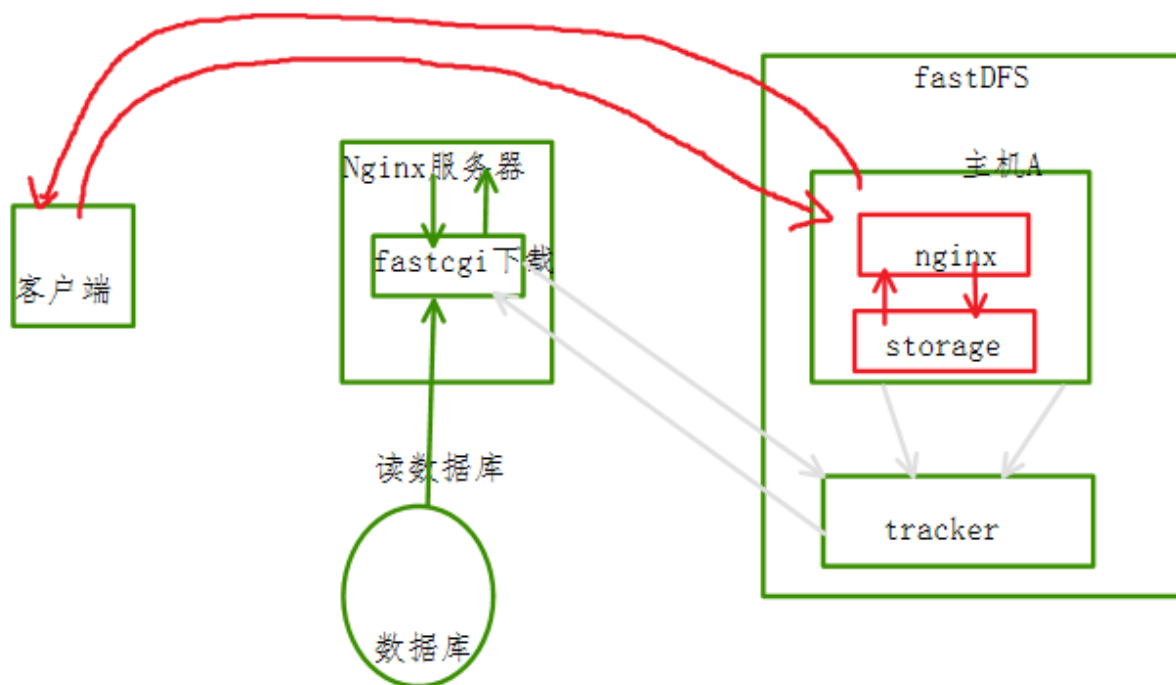
3. 优化

优化思路:

- 直接让客户端连接fastDFS的存储节点, 实现文件下载

- 举例, 访问一个url直接下载:

http://192.168.247.147/group1/M00/00/00/wKj3k1tMBKuARhwBAAvea_OGt2M471.jpg



1. 客户端发送请求使用的协议: http
 - fastDFS能不能解析http协议
 - nginx能解析http协议
 - 在nginx中安装fastDFS的插件
2. 客户端怎么知道文件就存储在对应的那个存储节点上?
 - 上传的时候将fileID和存储节点IP地址都进行存储

2. Nginx和fastDFS的整合

1. 在存储节点上安装Nginx, 将软件安装包拷贝到fastDFS存储节点对应的主机上

```
1 # 1. 找fastDFS的存储节点
2 # 2. 在存储节点对应的主机上安装Nginx, 安装的时候需要一并将插件装上
3 # - (余庆提供插件的代码 + nginx的源代码) * 交叉编译 = Nginx
```

2. 在存储节点对应的主机上安装Nginx, 作为web服务器

```

1 - fastdfs-nginx-module_v1.16.tar.gz 解压缩
2 # 1. 进入nginx的源码安装目录
3 # 2. 检测环境, 生成makefile
4 ./configure --add-module=fastdfs插件的源码目录/src
5 make
6 sudo make install

```

make过程中的错误:

```

1 # 1. fatal error: fdfs_define.h: No such file or directory
2 # 2. fatal error: common_define.h: No such file or directory
3
4 default:    build
5
6 clean:
7     rm -rf Makefile objs
8
9 build:
10     $(MAKE) -f objs/Makefile
11
12 install:
13     $(MAKE) -f objs/Makefile install
14
15 modules:
16     $(MAKE) -f objs/Makefile modules
17
18 upgrade:
19     /usr/local/nginx/sbin/nginx -t
20
21     kill -USR2 `cat /usr/local/nginx/logs/nginx.pid`
22     sleep 1
23     test -f /usr/local/nginx/logs/nginx.pid.oldbin
24
25     kill -QUIT `cat /usr/local/nginx/logs/nginx.pid.oldbin`
26 # 解决方案 - 修改objs/Makefile
27 ALL_INCS = -I src/core \
28             -I src/event \
29             -I src/event/modules \
30             -I src/os/unix \
31             -I /usr/local/include/fastdfs \
32             -I /usr/local/include/fastcommon/ \
33             -I objs \
34             -I src/http \
35
36             -I src/http/modules\
37             -I /usr/include/fastdfs/

```

3. 安装成功,启动Nginx,发现没有 worker进程

```

1  robin@OS:/usr/local/nginx/sbin$ ps aux|grep nginx
2  root      65111  0.0  0.0 39200  696 ?Ss   10:32   0:00 nginx: master process
   ./nginx
3  robin     65114  0.0  0.0 16272  928 pts/9  S+   10:32   0:00 grep --color=auto
   nginx

```

找nginx的logs日志

```

1  # ERROR - file: shared_func.c, line: 968, file /etc/fdfs/mod_fastdfs.conf not exist
2  # 从哪儿找 -> fastDFS插件目录中找
3  robin@OS:~/package/nginx/fastdfs-nginx-module/src$ tree
4  .
5  ├── common.c
6  ├── common.h
7  ├── config
8  ├── mod_fastdfs.conf    -> cp /etc/fdfs
9  └── ngx_http_fastdfs_module.c
10
11  0 directories, 5 files

```

需要修改mod_fdfs.conf文件, 参数当前存储节点的storage.conf进行修改

```

1  # 存储log日志的目录
2  base_path=/home/robin/fastdfs/storage
3  # 连接tracker地址信息
4  tracker_server=192.168.247.135:22122
5  # 存储节点绑定的端口
6  storage_server_port=23000
7  # 当前存储节点所属的组
8  group_name=group1
9  # 客户端下载文件的时候, 这个下载的url中是不是包含组的名字
10 // 上传的fileID: group1/M00/00/00/wKj3h1vJRPeAA9KEAAIZMjR0rI076.cpp
11 // 完整的url: http://192.168.1.100/group1/M00/00/00/wKj3h1vJRPeAA9KEAAIZMjR0rI076.cpp
12 url_have_group_name = true
13 # 存储节点上存储路径的个数
14 store_path_count=1
15 # 存储路径的详细信息
16 store_path0=/home/robin/fastdfs/storage

```

4. 重写启动Nginx, 还是没有worker进程, 查看log日志

```

1  # ERROR - file: ini_file_reader.c, line: 631, include file "http.conf" not exists,
   line: "#include http.conf"
2  从 /etc/fdfs 下找的时候不存在
3      - 从fastDFS源码安装目录找/conf
4      - sudo cp http.conf /etc/fdfs
5  # ERROR - file: shared_func.c, line: 968, file /etc/fdfs/mime.types not exist
6      - 从nginx的源码安装包中找/conf
7      - sudo cp mime.types /etc/fdfs

```

5. 通过浏览器请求服务器下载文件: 404 Not Found

```

1 http://192.168.1.100/group1/M00/00/00/wKj3h1vJRPAA9KEAAAIzMjR0rI076.jpg
2 # 错误信息
3 open() "/usr/local/nginx/zyFile2/group1/M00/00/00/wKj3h1vJS0qAM6RHAAvqH_kipG8229.jpg"
  failed (2: No such file or directory), client: 192.168.247.1, server: localhost,
  request: "GET /group1/M00/00/00/wKj3h1vJS0qAM6RHAAvqH_kipG8229.jpg HTTP/1.1", host:
  "192.168.247.135"
4 服务器在查找资源时候，找的位置不对，需要给服务器指定一个正确的位置，如何指定？
5     - 资源在哪？在存储节点的存储目录中 store_path0
6     - 如何告诉服务器资源在这？在服务器端添加location处理
7     locatioin /group1/M00/00/00/wKj3h1vJS0qAM6RHAAvqH_kipG8229.jpg
8     location /group1/M00/00/00/
9     location /group1/M00/
10    location /group1/M00/
11    {
12        # 告诉服务器资源的位置
13        root /home/robin/fastdfs/storage/data;
14        ngx_fastdfs_module;
15    }
16

```

3. 数据库表

3.1 数据库操作

#

1. 创建一个名称为cloud_disk的数据库

```
1 CREATE DATABASE cloud_disk;
```

2. 删除数据库cloud_disk

```
1 drop database cloud_disk;
```

3. 使用数据库 cloud_disk

```
1 use cloud_disk;
```

3.2 数据库建表

#

1. 用户信息表 -- user

| 字段 | 解释 |
|------------|----------------|
| id | 用户序号, 自动递增, 主键 |
| name | 用户名字 |
| nickname | 用户昵称 |
| phone | 手机号码 |
| email | 邮箱 |
| password | 密码 |
| createtime | 时间 |

```

1  CREATE TABLE user (
2      id BIGINT NOT NULL PRIMARY KEY AUTO_INCREMENT,
3      name VARCHAR (128) NOT NULL,
4      nickname VARCHAR (128) NOT NULL,
5      password VARCHAR (128) NOT NULL,
6      phone VARCHAR (15) NOT NULL,
7      createtime VARCHAR (128),
8      email VARCHAR (100),
9      CONSTRAINT uq_nickname UNIQUE (nickname),
10     CONSTRAINT uq_name UNIQUE (NAME)
11 );

```

2. 文件信息表 - user_file_list

| 字段 | 解释 |
|----------|---|
| md5 | 文件md5, 识别文件的唯一表示(身份证号) |
| file_id | 文件id-/group1/M00/00/00/xxx.png |
| url | 文件url 192.168.1.2:80/group1/M00/00/00/xxx.png - 下载的时候使用 |
| size | 文件大小, 以字节为单位 |
| type | 文件类型: png, zip, mp4..... |
| fileName | 文件名 |
| count | 文件引用计数, 默认为1 每增加一个用户拥有此文件, 此计数器+1 |

```

1  CREATE TABLE user_file_list (
2      user VARCHAR (128) NOT NULL,
3      md5 VARCHAR (200) NOT NULL,
4      createtime VARCHAR (128),
5      filename VARCHAR (128),
6      shared_status INT,
7      pv INT
8  );

```

3. 用户文件列表 - user_file_list

| 字段 | 解释 |
|---------------|----------------------|
| user | 文件所属用户 |
| md5 | 文件md5 |
| createtime | 文件创建时间 |
| filename | 文件名字 |
| shared_status | 共享状态, 0为没有共享, 1为共享 |
| pv | 文件下载量, 默认值为0, 下载一次加1 |

```
1 CREATE TABLE user_file_list (  
2     user VARCHAR (128) NOT NULL,  
3     md5 VARCHAR (200) NOT NULL,  
4     createtime VARCHAR (128),  
5     filename VARCHAR (128),  
6     shared_status INT,  
7     pv INT  
8 );
```

4. 用户文件数量表 - user_file_count

| 字段 | 解释 |
|-------|---------|
| user | 文件所属用户 |
| count | 拥有文件的数量 |

```
1 CREATE TABLE user_file_count (  
2     user VARCHAR (128) NOT NULL PRIMARY KEY,  
3     count INT  
4 );
```

5. 共享文件列表 - share_file_list

| 字段 | 解释 |
|------------|----------------------|
| user | 文件所属用户 |
| md5 | 文件md5 |
| createtime | 文件共享时间 |
| filename | 文件名字 |
| pv | 文件下载量, 默认值为1, 下载一次加1 |

```

1 CREATE TABLE share_file_list (
2     user VARCHAR (128) NOT NULL,
3     md5 VARCHAR (200) NOT NULL,
4     createtime VARCHAR (128),
5     filename VARCHAR (128),
6     pv INT
7 );

```

复习

1. fastCGI

1. 是什么?

- 运行在服务器端的代码, 帮助服务器处理客户端提交的动态请求

2. 干什么

- 帮助服务器处理客户端提交的动态请求

3. 怎么用?

- nginx如何转发数据

```

1 # 分析出客户端请求对应的指令 -- /test
2 location /test
3 {
4     # 转发出去
5     fastcgi_pass 地址:端口;
6     include fastcgi.conf;
7 }

```

- fastcgi如何接收数据

```

1 # 启动, 通过spawn-fcgi启动
2 spawn-fcgi -a IP -p port -f ./fcgi
3 # 编写fastCGI程序的时候
4 - 接收数据: 调用读终端的函数就是接收数据
5 - 发送数据: 调用写终端的函数就是发送数据

```

- fastcgi如何处理数据

```

1 // 编写登录的fastCgi程序
2 int main()
3 {
4     while(FCGI_Accept() >= 0)
5     {
6         // 1. 接收登录信息 -> 环境变量中
7         // post -> 读数据块的长度 CONTENT_LENGTH
8         // get -> 从请求行的第二部分读 QUERY_STRING
9         // 2. 处理数据
10        // 3. 回发结果 -> 格式假设是json
11        printf("Content-type: application/json");
12        printf("{\"status\":\"OK\"}");
13    }

```