

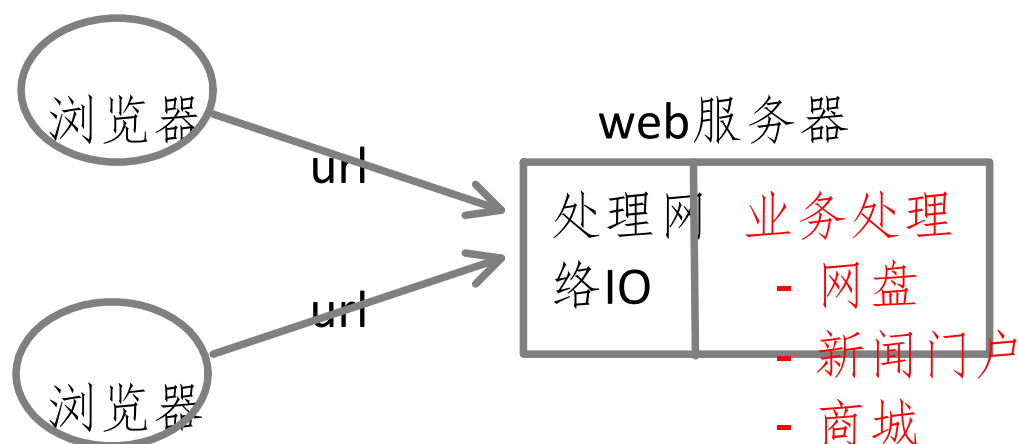
xx.avi

1. 通过浏览器/桌面客户端访问服务器
 - c/s
 - b/s
2. 反向代理服务器
 - 多台web服务器 - 集群
 - 给web服务器分配资源
3. 高并发
 - 多台web服务器
4. nginx服务器+fastcgi
 - nginx 处理静态请求
 - 动态需要使用FASTCGI处理
5. 数据库
 - mysql
 - redis
6. 分布式的文件系统 - fastDFS
 - 上传和下载文件

Web服务器

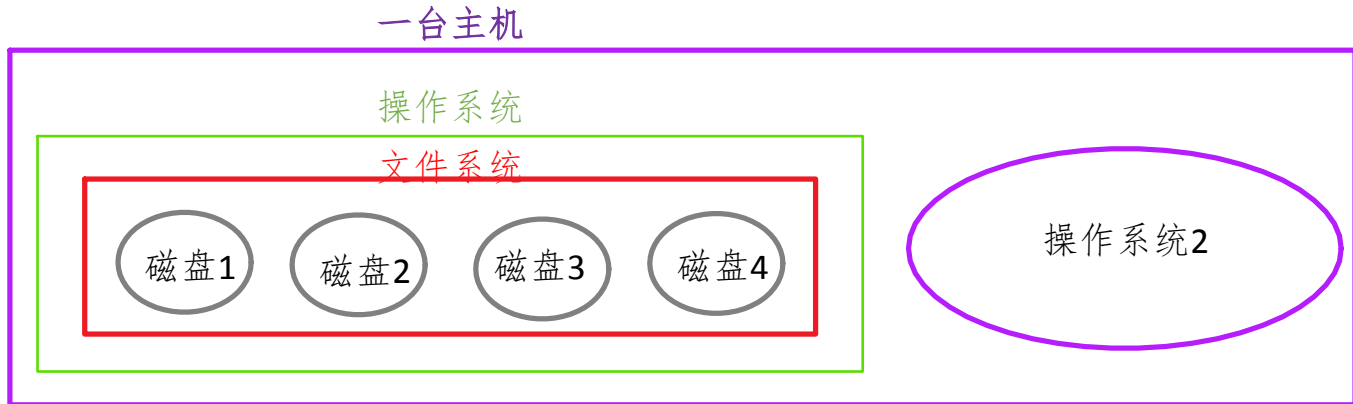
1. 什么是服务器
 - 硬件:
 - 一台配置比较高的电脑
 - 软件:
 - 在电脑上安装服务器软件
2. 常见的web服务器
 - tomcat服务器
 - apache组织产品, 开源的免费服务器
 - weblogic服务器
 - bea公司, 收费的服务器
 - 不交费, 访问量受限制
 - IIS服务器
 - Internet Information Server
 - 微软公司主推的服务器
 - nginx
 - 小巧且高效的HTTP服务器
 - 也可以做一个高效的负载均衡反向代理

web服务器 - 能解析http协议的服务器

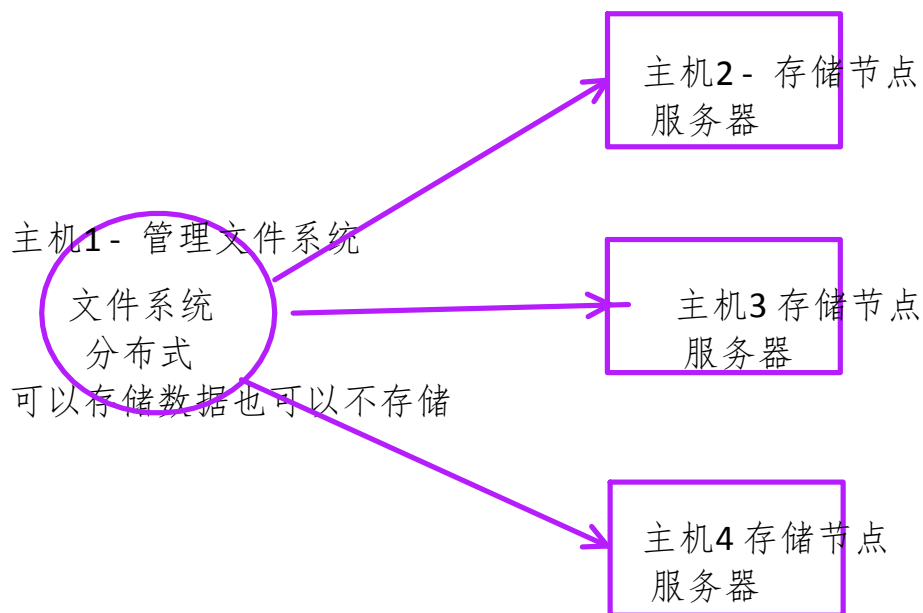


分布式文件系统

1. 文件系统 - 存储数据
fat32, ntfs, ext3, ext4
2. 分布式
 - 一般文件系统



- 分布式的文件系统
 - 文件系统的全部，不在同一台主机上，在很多台主机上，多个分散的文件系统组合在一起，形成了一个完整的文件系统。
 - 分布式的文件系统基本结构



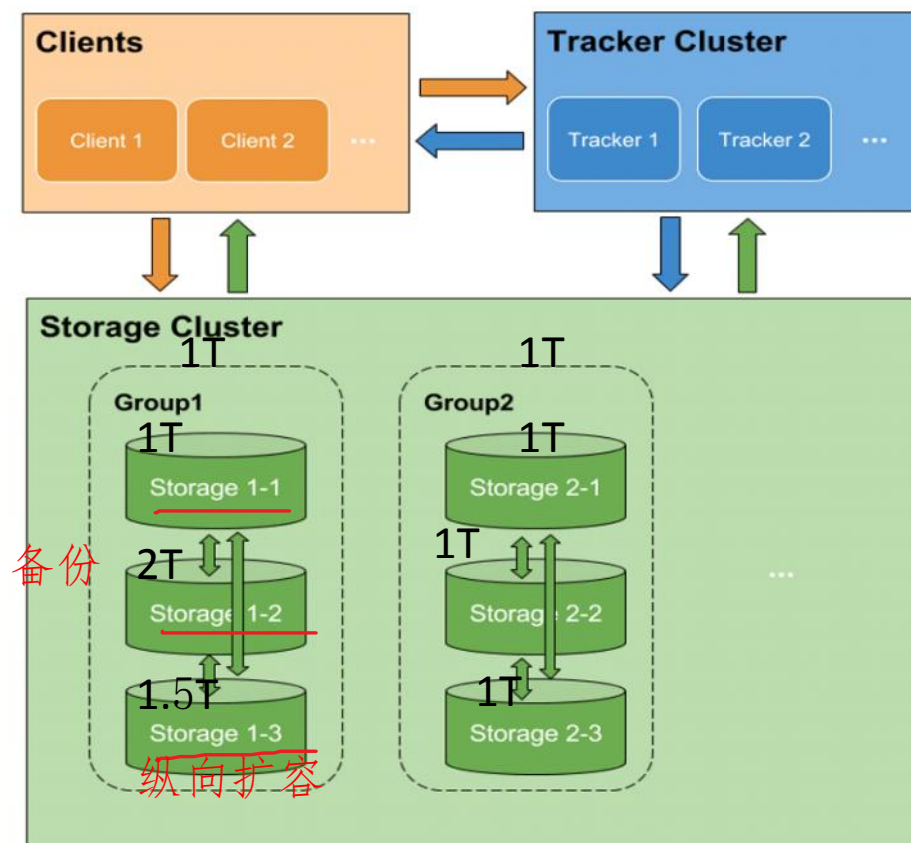
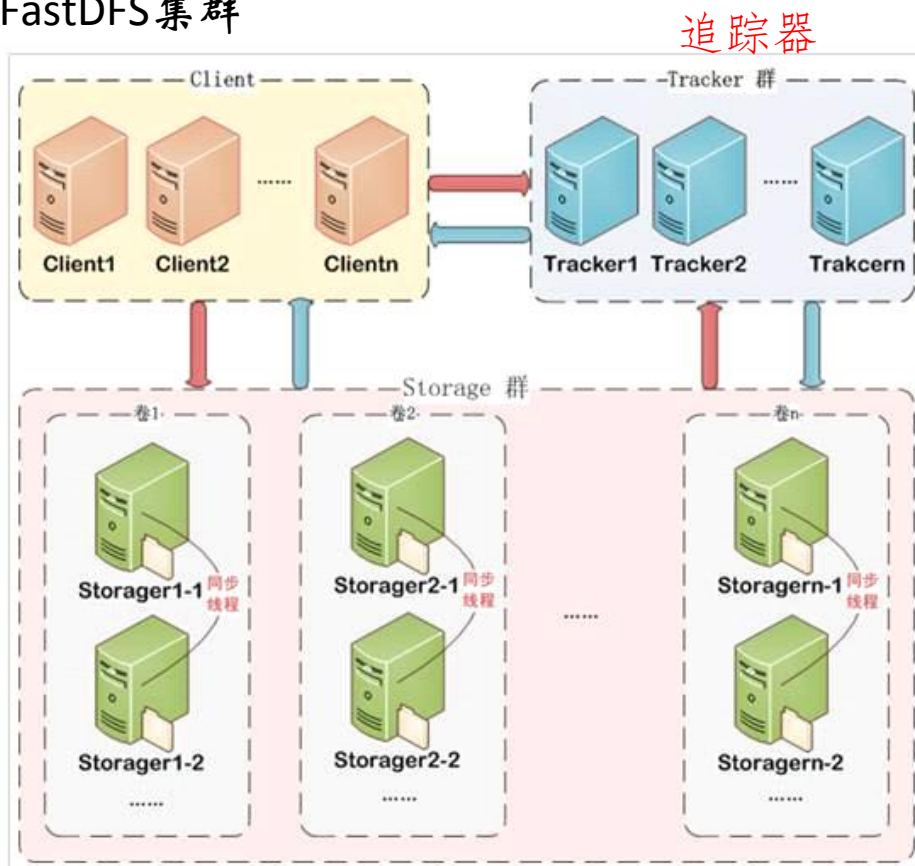
fastDFS实现分布式文件系统的搭建



fastDFS整个分布式文件系统

FastDFS 集群 - (了解内容)

FastDFS 集群



1. Tracker集群

- Tracker server之间是相互平等关系同时提供服务
- Tracker server不存在单点故障。
- 客户端请求Tracker server采用轮询方式，如果请求的tracker无法提供服务则换另一个tracker。

2. Storage集群

- Storage集群采用了分组存储方式，由一个或多个组构成
- 集群存储总容量为集群中所有组的存储容量之和
- 一个组由一台或多台存储服务器组成，组内的Storage server之间是平等关系
- 不同组的Storage server之间不会相互通信，同组内的Storage server之间会相互连接进行文件同步，从而保证同组内每个storage上的文件完全一致的。
- 一个组的存储容量为该组内存储服务器容量最小的那个

3. FastDFS的扩容分纵向扩容与横向扩容

- 横向扩容
 - 增加容量
 - 添加group组
- 纵向扩容
 - 数据备份
 - 当前组的最大容量
 - 存储节点中容量最小的那个
 - 所有存储节点组名必须一样

fastDFS 安装

1. 安装

- 基础库包

- libfastcommon-1.36.zip
 - ./make.sh
 - sudo ./make.sh install

- fastDFS

- fastdfs-5.10.tar.gz
 - ./make.sh
 - sudo ./make.sh install

2. 测试是否安装成功

- fdfs_test

找不到动态库.so的问题

第一种：

- 使用环境变量 `LD_LIBRARY_PATH`
- 动态库的绝对路径添加到该环境变量中
- `LD_LIBRARY_PATH = /usr/include/xxx:$LD_LIBRARY_PATH`
- `export LD_LIBRARY_PATH`
 - 直接在shell中 - 临时设置
 - 写入配置文件
 - `~/.bashrc` - 用户级别
 - `/etc/profile` - 系统级别

第二种：

- 给动态库设置软连接 `/usr/kevin/fastdfs`
- 将软链接放到 `/usr/lib` `/usr/lib64` `/lib` `/lib64`
- `sudo ln -s /usr/kevin/fastdfs/xxx.so /usr/lib/libxxx.so`

第三种：

- 刷新 `/etc/ld.so.cache`
 - 首先 `vi /etc/ld.so.conf`
 - 将动态库的绝对路径写入该文件中
 - `sudo ldconfig -v`

fastDFS配置

1. 默认配置文件位置：

- /etc/fdfs

2. Tracker（追踪器）

- 修改配置文件

- bind_addr=192.168.82.253
 - 当前主机的IP
- port=22122
 - 绑定的端口
- base_path=/home/robin/fastDFS/tracker
 - log目录

- 启动Tracker服务

- fdfs_trackerd /etc/fdfs/tracker.conf
- sudo fdfs_trackerd /etc/fdfs/tracker.conf restart
- sudo fdfs_trackerd /etc/fdfs/tracker.conf stop

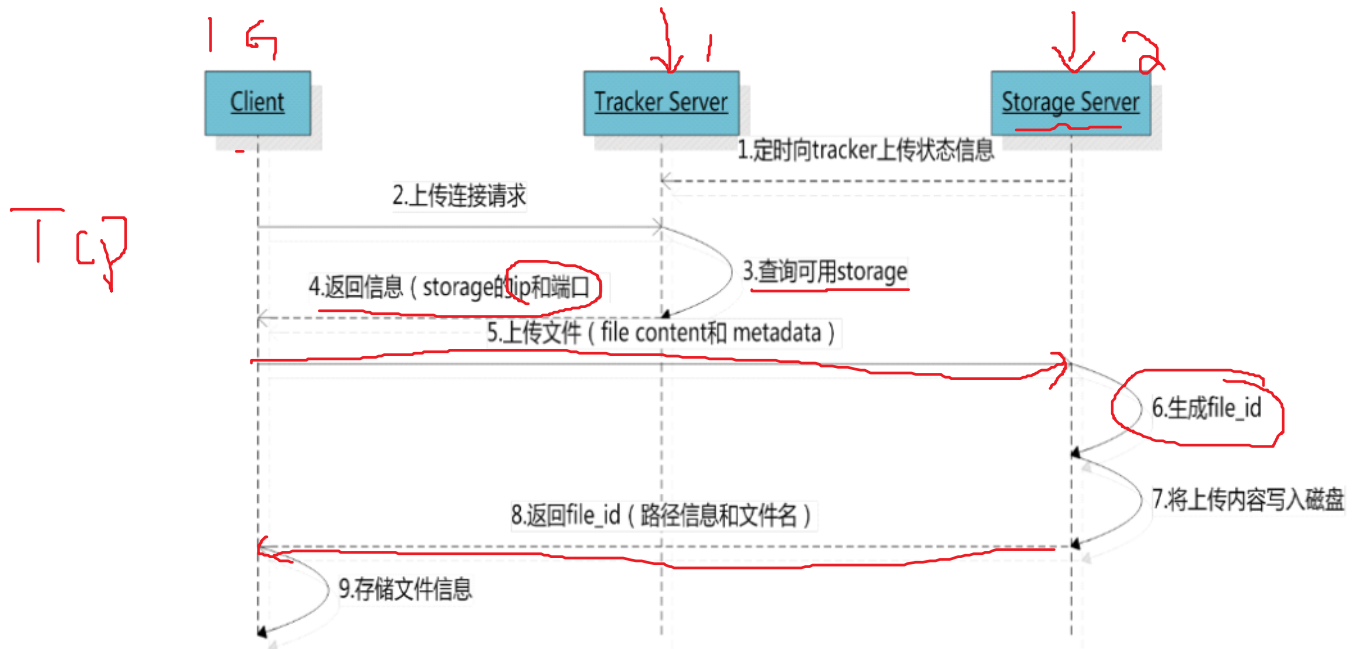
3. Storage（存储节点）

- 修改配置文件

- group_name=group1
 - 存储节点所属的组
- bind_addr=192.168.82.253
 - 存储节点绑定的IP
- port=23000
 - 绑定的端口
- base_path=/home/robin/fastDFS/storage
 - 存储日志文件的目录
- store_path_count=1
 - 存储目录的个数
- store_path0=/home/robin/fastDFS/storage

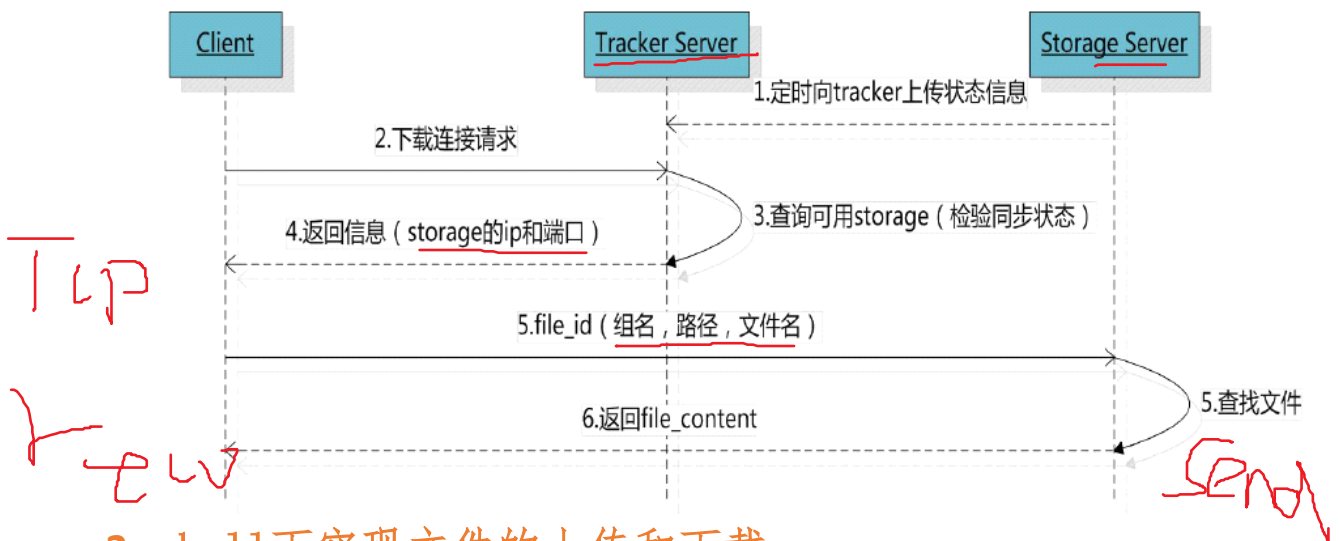
- 配置具体的存储目录
 - tracker_server=192.168.82.253:22122
 - 连接tracker的时候使用的IP和端口
- 启动Storage服务
 - fdfs_storaged /etc/fdfs/storage.conf
 - sudo fdfs_storaged /etc/fdfs/storage.conf restart
 - sudo fdfs_storaged /etc/fdfs/storage.conf stop
- 4. Client
 - 修改配置文件
 - base_path=/home/robin/fastDFS/client
 - log日志目录
 - tracker_server=192.168.82.253:22122
 - 连接tracker时候需要的IP和端口信息
- 5. 检测当前fdfs环境是否是通的
 - fdfs_monitor /etc/fdfs/client.conf
 - ACITVE

1. 文件上传



存储文件信息需要程序猿去处理

2. 文件下载



3. shell下实现文件的上传和下载

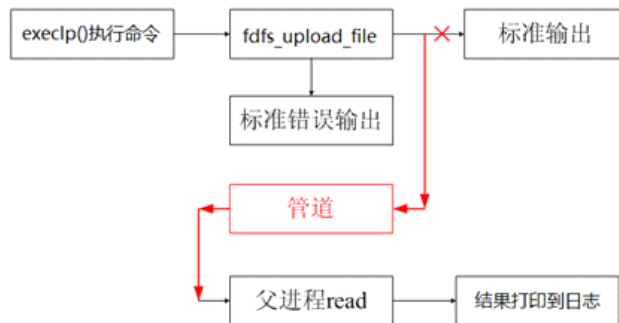
README.MD ==> [group1/M00/00/00/wKhS_VlrEf0AdIZyAAAJT0wCGr43848.md](#)

- group1 - 组名
- M00 - 虚拟磁盘路径
 - 对应storage.conf - store_path*=xxxxx
- 00/00 - 二级目录, 存储上传文件的目录
- wKhS_VlrEf0AdIZyAAAJT0wCGr43848.md - 文件名
 - 文件名包含的信息
 - 采用Base64编码
 - 包含的字段包括
 - 源storage server Ip 地址

- 文件创建时间
- 文件大小
- 文件CRC32效验码
 - ▶ 循环冗余校验
- 随机数

代码实现fastdfs文件上传和下载

1. fastDFS api实现
2. 进程方式实现
 - 创建匿名管道
 - 创建子进程
 - 子进程中执行execvp
 - 关闭读端
 - 将得到的文件ID写入管道
 - 父进程
 - 读管道
 - 关闭写端
 - 将fileID读到内存
 - 回收子进程pcb



execvp()
execvp
/usr/bin

1. 父子进程始终共享的东西
 - 文件描述符
 - open
 - pipe
 - 内存映射区 - mmap

数据库和文件系统

2017年7月16日 10:09

