

# FastDFS

---

需求：搭建一个单独的图片服务器。

## 为什么要使用分布式文件系统？

---

### 技术演进

#### 1. 单体架构

直接在项目中创建静态资源文件夹，使用相对路径或者绝对路径来获取图片等资源。

**优点：**方便快捷，实现简单

**缺点：**资源文件越来越大，打包越来越大，不利于解耦

#### 2. 独立文件服务器

分离图片等静态资源文件和代码的耦合度。通过ftp或者ssh将图片上传到一个单独的服务器的目录下，再通过nginx或apache来访问此目录下的文件，返回一个独立的图片URL地址

**优点：**将需要大量网络IO的图片访问分离出来，避免消耗大量的服务器资源，独立储存可以动态扩容，方便迁移。

**缺点：**单机情况下存在性能瓶颈，需要手动实现容灾等

#### 3. 分布式文件系统

独立服务器可以降低对项目本体的资源消耗，但是会存在容灾等问题，单机也会有性能瓶颈。这时候就需要分布式文件系统。分布式文件系统一般会具有3个角色，分别负责储存、访问的目录系统，真正负责储存的储存系统，容灾系统来构成。

**优点：**强大的扩展能力，高可用

**缺点：**系统复杂度加大，需要更多服务器部署

## FastDFS

---

### 什么是FastDFS

FastDFS是一个开源的轻量级分布式文件系统。它解决了大数据量存储和负载均衡等问题。特别适合以中小文件（建议范围：4KB < file\_size < 500MB）为载体的在线服务，如相册网站、视频网站等等。在UC基于FastDFS开发向用户提供了：网盘，社区，广告和应用下载等业务的存储服务。

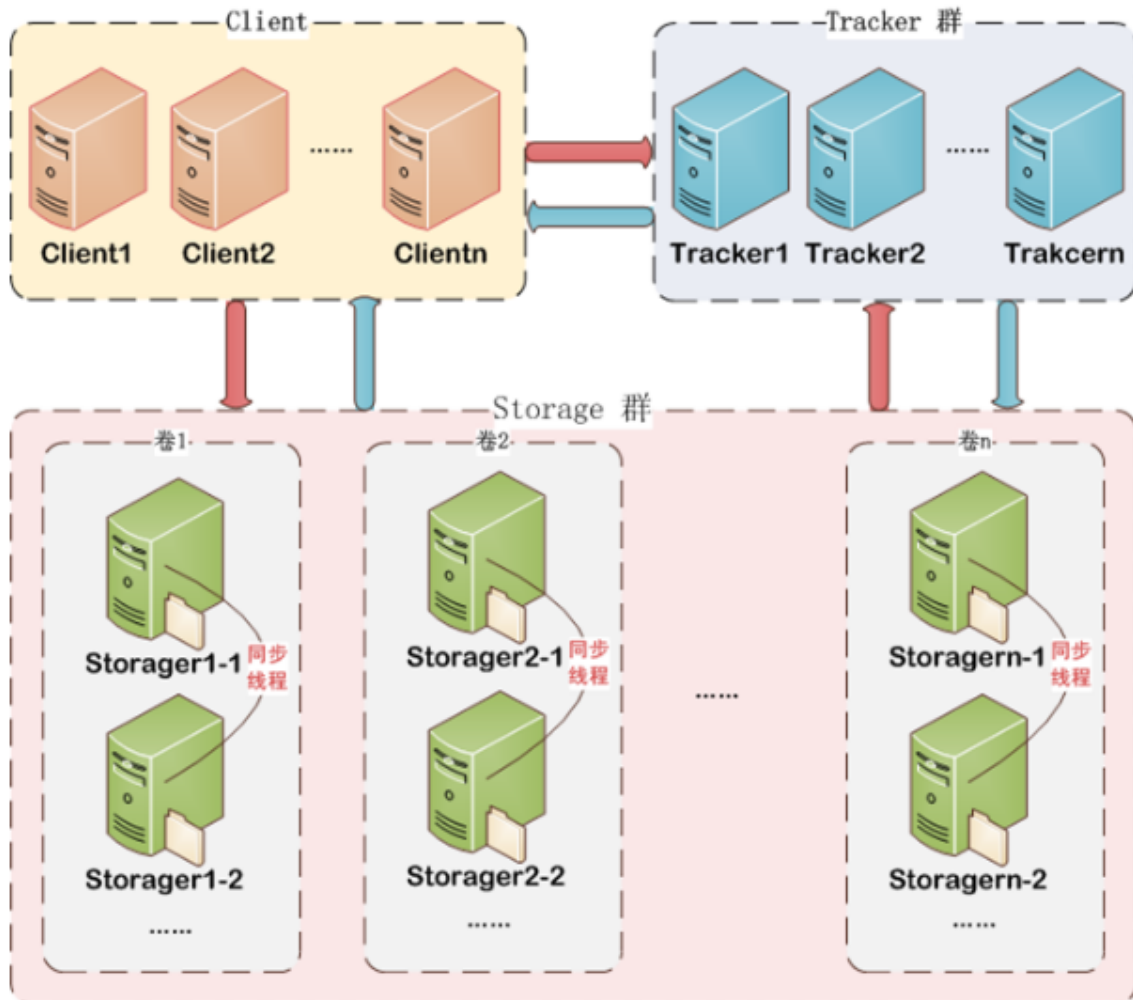
FastDFS是一款开源的轻量级分布式文件系统纯C实现，支持Linux、FreeBSD等UNIX系统类google FS，不是通用的文件系统，只能通过专有API访问，目前提供了C、Java和PHP API为互联网应用量身定做，解决大容量文件存储问题，追求高性能和高扩展性FastDFS可以看做是基于文件的key value pair存储系统，称作分布式文件存储服务更为合适。

### FastDFS特性

- 文件不分块存储，上传的文件和OS文件系统中的文件——对应
- 支持相同内容的文件只保存一份，节约磁盘空间
- 下载文件支持HTTP协议，可以使用内置Web Server，也可以和其他Web Server配合使用
- 支持在线扩容
- 支持主从文件
- 存储服务器上可以保存文件属性（meta-data） V2.0网络通信采用libevent，支持大并发访问，整体性能更好

## FastDFS相关概念

FastDFS服务端有三个角色：跟踪服务器（tracker server）、存储服务器（storage server）和客户端（client）。



跟踪服务器：

储存服务器：

客户端：

group：

meta data：

**上传机制**

**下载机制**

**同步时间**

**文件ID-FID**

# 搭建FastDFS

## SpringBoot整合FastDFS

### 1. pom

```
<dependency>
  <groupId>org.csource</groupId>
  <artifactId>fastdfs-client-java</artifactId>
  <version>1.29-SNAPSHOT</version>
</dependency>
```

**ps:**

fastdfs-client并没有发布到Maven中央仓库，需要自己下载源码编译后打包

源码地址 <https://github.com/happyfish100/fastdfs-client-java>

下载后执行 `mvn clean install` (需要Maven环境)

目前最新包为1.29，其中有少部分方法和1.27不一致

如：1.27中 `trackerClient.getConnection()`

1.29为 `trackerClient.getTrackerServer();`

使用时注意修改

### 2. 配置文件

resource目录下添加 `fdfs_client.conf` 文件

```
connect_timeout = 2
network_timeout = 30
charset = UTF-8
http.tracker_http_port = 80
http.anti_steal_token = no
http.secret_key = FastDFS1234567890

tracker_server = 10.0.11.247:22122
tracker_server = 10.0.11.248:22122
tracker_server = 10.0.11.249:22122

connection_pool.enabled = true
connection_pool.max_count_per_entry = 500
connection_pool.max_idle_time = 3600
connection_pool.max_wait_time_in_ms = 1000
```

注1: `tracker_server`指向您自己IP地址和端口，1-n个

注2: 除了`tracker_server`，其它配置项都是可选的

详细内容参考：

### 3.

## 拾遗