

bojiangzhou

英雄修身齐家治国平天下

公告

昵称: bojiangzhou
园龄: 2年7个月
粉丝: 195
关注: 7
+加关注

<	2018年7月						>
日	一	二	三	四	五	六	
24	25	26	27	28	29	30	
1	2	3	4	5	6	7	
8	9	10	11	12	13	14	
15	16	17	18	19	20	21	
22	23	24	25	26	27	28	
29	30	31	1	2	3	4	

最新随笔

1. 服务器集成环境搭建
2. 多租户个性化服务路由
3. 基于SpringBoot搭建应用开发框架(一) —— 基础架构
4. Redis 学习(三) —— 事务、消息发布订阅
5. Redis 学习(二) —— 数据类型及操作
6. Redis 学习(一) —— 安装、通用key操作命令
7. 用FastDFS一步步搭建文件管理系统
8. 在IntelliJ IDEA中使用Debug
9. 毕业设计之进销存管理系统 —— 一步步搭建自己的框架及系统
10. 版本管理工具 —— SVN

随笔分类(39)

- dbutils(1)
- easyui(2)
- ExtJs(1)
- fastdfs(1)
- hibernate(1)
- IDE(1)
- java(5)
- JavaScript(6)
- mybatis(1)
- mysql(7)
- redis(3)
- spring(1)
- spring boot(1)
- struts
- 环境搭建(1)
- 框架
- 设计模式(1)
- 项目(4)

博客园 首页 新随笔 联系 管理

随笔 - 38 文章 - 12 评论 - 240

用FastDFS一步步搭建文件管理系统

目录

- 一、FastDFS介绍
 - 1、简介
 - 2、FastDFS的存储策略
 - 3、FastDFS的上传过程
 - 4、FastDFS的文件同步
 - 5、FastDFS的文件下载
- 二、安装FastDFS环境
 - 0、前言
 - 1、下载安装 libfastcommon
 - 2、下载安装FastDFS
 - 3、配置FastDFS跟踪器(Tracker)
 - 4、配置 FastDFS 存储 (Storage)
 - 5、文件上传测试
- 三、安装Nginx
 - 1、安装nginx所需环境
 - 2、安装Nginx
 - 3、访问文件
- 四、FastDFS 配置 Nginx 模块
 - 1、安装配置Nginx模块
- 五、Java客户端
 - 1、首先需要搭建 FastDFS 客户端Java开发环境
 - 2、客户端API
 - 六、权限控制

[回到顶部](#)

一、FastDFS介绍

FastDFS开源地址: <https://github.com/happyfish100>

参考: [分布式文件系统FastDFS设计原理](#)

参考: [FastDFS分布式文件系统](#)

个人封装的FastDFS Java API: <https://github.com/bojiangzhou/lyyzoo-fastdfs-java>

1、简介

FastDFS 是一个开源的高性能分布式文件系统 (DFS) 。它的主要功能包括: 文件存储, 文件同步和文件访问, 以及高容量和负载均衡。主要解决了海量数据存储问题, 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。

FastDFS 系统有三个角色: 跟踪服务器(Tracker Server)、存储服务器(Storage Server)和客户端(Client)。

Tracker Server: 跟踪服务器, 主要做调度工作, 起到均衡的作用; 负责管理所有的 storage server和 group, 每个storage 在启动后会连接 Tracker, 告知自己所属 group 等信息, 并保持周期性心跳。

Storage Server: 存储服务器, 主要提供容量和备份服务; 以 group 为单位, 每个 group 内可以有多台 storage server, 数据互为备份。

Client: 客户端, 上传下载数据的服务器, 也就是我们自己的项目所部署在的服务器。

项目管理(2)

随笔档案(38)

2018年6月 (1)

2018年5月 (1)

2018年2月 (1)

2017年11月 (2)

2017年10月 (2)

2017年9月 (1)

2017年6月 (1)

2017年3月 (1)

2017年2月 (1)

2017年1月 (1)

2016年10月 (3)

2016年9月 (4)

2016年8月 (1)

2016年7月 (3)

2016年6月 (2)

2016年5月 (2)

2016年3月 (1)

2016年2月 (2)

2016年1月 (1)

2015年12月 (7)

积分与排名

积分 - 86826

排名 - 4250

最新评论

1. Re:学生成绩管理系统/学生信息管理系统

楼主，能不能加个好友，我总是报404错误，我把这个弄成了课程设计，但是我总是出现HTTP Status 404 - /ssms/index.jsptype Status reportmessage /.....

--呆呆的小猪

2. Re:基于SpringBoot搭建应用开发框架(一) —— 基础架构

有几处写错了或者没写清楚：1. Result类，需要依赖jackson-annotations2. UserController.queryAll()返回值应是：... BaseEnums.SUCCE.....

--Von.Schütze

3. Re:jQuery jsonp跨域请求

写的很详细，楼主，谢谢分享；

--Jsk201

4. Re:多租户个性化服务路由

优秀，游戏。很优雅。

--demoofbug

5. Re:jQuery jsonp跨域请求可以的

--妙人

阅读排行榜

1. jQuery jsonp跨域请求(71549)

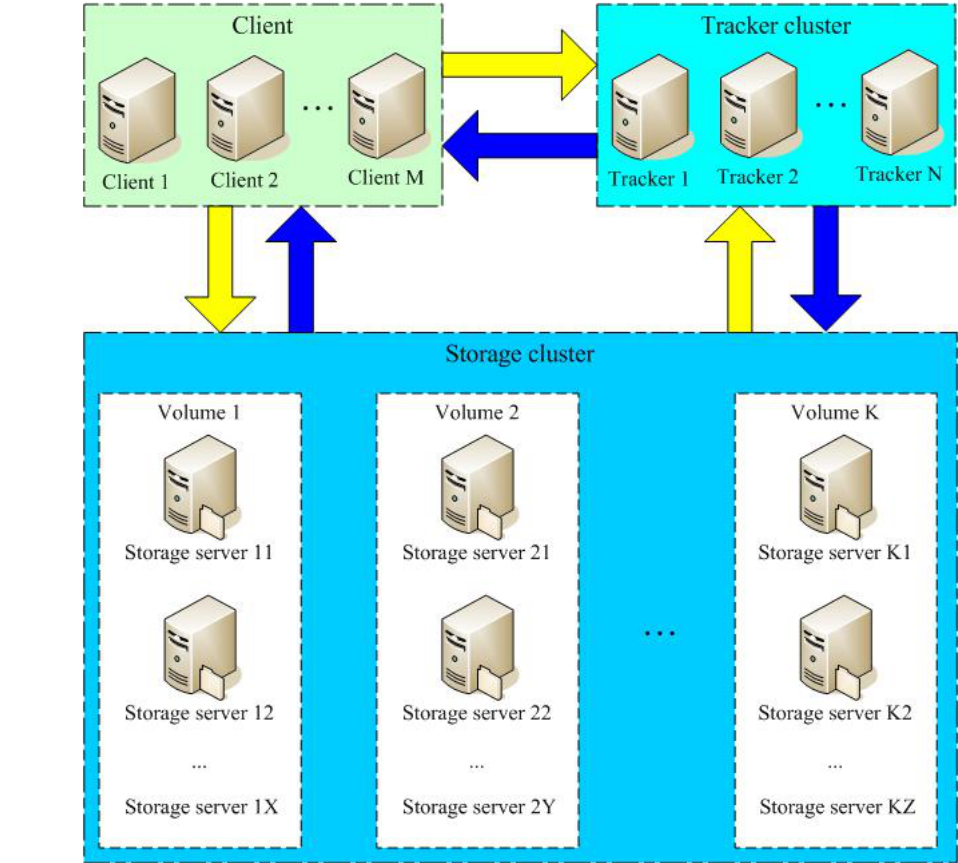
2. easyui-textbox 和 easyui-validatebox 设置值和获取值(42404)

3. 在IntelliJ IDEA中使用Debug(34003)

4. 学生成绩管理系统/学生信息管理系统(31770)

5. 用FastDFS一步步搭建文件管理系统(30810)

评论排行榜



2、FastDFS的存储策略

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。

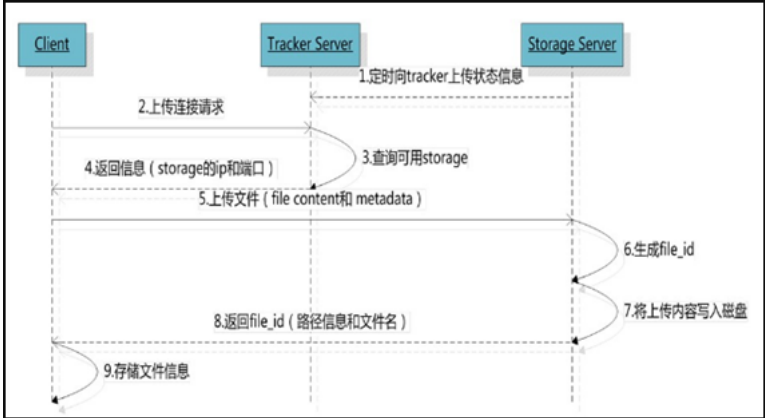
在卷中增加服务器时，同步已有的文件由系统自动完成，同步完成后，系统自动将新增服务器切换到线上提供服务。当存储空间不足或即将耗尽时，可以动态添加卷。只需要增加一台或多台服务器，并将它们配置为一个新的卷，这样就扩大了存储系统的容量。

3、FastDFS的上传过程

FastDFS向使用者提供基本文件访问接口，比如upload、download、append、delete等，以客户端库的方式提供给用户使用。

Storage Server会定期的向Tracker Server发送自己的存储信息。当Tracker Server Cluster中的Tracker Server不止一个时，各个Tracker之间的关系是对等的，所以客户端上传时可以选择任意一个Tracker。

当Tracker收到客户端上传文件的请求时，会为该文件分配一个可以存储文件的group，当选定了group后就要决定给客户端分配group中的哪一个storage server。当分配好storage server后，客户端向storage发送写文件请求，storage将会为文件分配一个数据存储空间。然后为文件分配一个fileid，最后根据以上的信息生成文件名存储文件。



4、FastDFS的文件同步

写文件时，客户端将文件写至group内一个storage server即认为写文件成功，storage server写完文件后，会由后台线程将文件同步至group内其他的storage server。

每个storage写文件后，同时会写一份binlog，binlog里不包含文件数据，只包含文件名等元信息，这份binlog用于后台同步，storage会记录向group内其他storage同步的进度，以便重启后能接上次的进度继续同步；进度以时间戳的方式进行

- 1. 毕业设计之进销存管理系统——一步步搭建自己的框架及系统(62)
- 2. 学生成绩管理系统/学生信息管理系统(42)
- 3. 基于SpringBoot搭建应用开发框架(一)——基础架构(39)
- 4. 用FastDFS一步步搭建文件管理系统(16)
- 5. 在IntelliJ IDEA中使用Debug(13)

推荐排行榜

- 1. 基于SpringBoot搭建应用开发框架(一)——基础架构(39)
- 2. 毕业设计之进销存管理系统——一步步搭建自己的框架及系统(37)
- 3. JQuery jsonp跨域请求(29)
- 4. 在IntelliJ IDEA中使用Debug(25)
- 5. 用FastDFS一步步搭建文件管理系统(9)

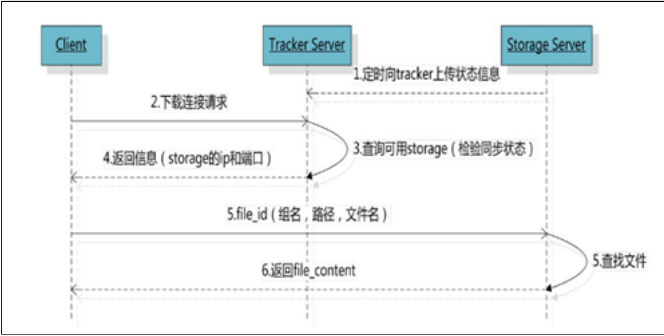
Copyright ©2018 bojiangzhou

记录，所以最好能保证集群内所有server的时钟保持同步。

storage的同步进度会作为元数据的一部分汇报到tracker上，tracker在选择读storage的时候会以同步进度作为参考。

5、FastDFS的文件下载

客户端uploadfile成功后，会拿到一个storage生成的文件名，接下来客户端根据这个文件名即可访问到该文件。



跟upload file一样，在downloadfile时客户端可以选择任意tracker server。tracker发送download请求给某个tracker，必须带上文件名信息，tracker从文件名中解析出文件的group、大小、创建时间等信息，然后为该请求选择一个storage用来服务读请求。

[回到顶部](#)

二、安装FastDFS环境

0、前言

操作环境：CentOS7 X64，以下操作都是单机环境。

我把所有的安装包下载到/softpackages/下，解压到当前目录。

先做一件事，修改hosts，将文件服务器的ip与域名映射(单机TrackerServer环境)，因为后面很多配置里面都需要去配置服务器地址，ip变了，就只需要修改hosts即可。

```
# vim /etc/hosts

增加如下一行，这是我的ip
192.168.51.128 file.ljzsg.com

如果要本机访问虚拟机，在C:\Windows\System32\drivers\etc\hosts中同样增加一行
```

1、下载安装 libfastcommon

libfastcommon是从 FastDFS 和 FastDHT 中提取出来的公共 C 函数库，基础环境，安装即可。

① 下载libfastcommon

```
# wget https://github.com/happyfish100/libfastcommon/archive/V1.0.7.tar.gz
```

② 解压

```
# tar -zxvf V1.0.7.tar.gz
# cd libfastcommon-1.0.7
```

③ 编译、安装

```
# ./make.sh
# ./make.sh install
```

④ libfastcommon.so 安装到了/usr/lib64/libfastcommon.so，但是FastDFS主程序设置的lib目录是/usr/local/lib，所以需要创建软链接。

```
# ln -s /usr/lib64/libfastcommon.so /usr/local/lib/libfastcommon.so
# ln -s /usr/lib64/libfastcommon.so /usr/lib/libfastcommon.so
# ln -s /usr/lib64/libfdfsclient.so /usr/local/lib/libfdfsclient.so
# ln -s /usr/lib64/libfdfsclient.so /usr/lib/libfdfsclient.so
```

2、下载安装FastDFS

① 下载FastDFS

```
# wget https://github.com/happyfish100/fastdfs/archive/V5.05.tar.gz
```

② 解压

```
# tar -zxvf V5.05.tar.gz
# cd fastdfs-5.05
```

③ 编译、安装

```
# ./make.sh
# ./make.sh install
```

④ 默认安装方式安装后的相应文件与目录

A、服务脚本：

```
/etc/init.d/fdfs_storaged
/etc/init.d/fdfs_trackerd
```

B、配置文件（这三个是作者给的样例配置文件）：

```
/etc/fdfs/client.conf.sample
/etc/fdfs/storage.conf.sample
/etc/fdfs/tracker.conf.sample
```

C、命令工具在 /usr/bin/ 目录下：

```
fdfs_appender_test
fdfs_appender_test1
fdfs_append_file
fdfs_crc32
fdfs_delete_file
fdfs_download_file
fdfs_file_info
fdfs_monitor
fdfs_storaged
fdfs_test
fdfs_test1
fdfs_trackerd
fdfs_upload_appender
fdfs_upload_file
stop.sh
restart.sh
```

⑤ FastDFS 服务脚本设置的 bin 目录是 /usr/local/bin，但实际命令安装在 /usr/bin/ 下。

两种方式：

》一是修改FastDFS 服务脚本中相应的命令路径，也就是把 /etc/init.d/fdfs_storaged 和 /etc/init.d/fdfs_trackerd 两个脚本中的 /usr/local/bin 修改成 /usr/bin。

```
# vim fdfs_trackerd
使用查找替换命令进统一修改:%s+/usr/local/bin+/usr/bin
# vim fdfs_storaged
使用查找替换命令进统一修改:%s+/usr/local/bin+/usr/bin
```

```
if [ ! -f /usr/local/bin/stop.sh ]; then
    echo "file /usr/local/bin/stop.sh does not exist!"
    exit 2
fi

if [ ! -f /usr/local/bin/restart.sh ]; then
    echo "file /usr/local/bin/restart.sh does not exist!"
    exit 2
fi
```

》二是建立 /usr/bin 到 /usr/local/bin 的软链接，我是用这种方式。

```
# ln -s /usr/bin/fdfs_trackerd /usr/local/bin
# ln -s /usr/bin/fdfs_storaged /usr/local/bin
# ln -s /usr/bin/stop.sh /usr/local/bin
# ln -s /usr/bin/restart.sh /usr/local/bin
```

3、配置FastDFS跟踪器(Tracker)

配置文件详细说明参考：[FastDFS 配置文件详解](#)

① 进入 /etc/fdfs，复制 FastDFS 跟踪器样例配置文件 tracker.conf.sample，并重命名为 tracker.conf。

```
# cd /etc/fdfs
# cp tracker.conf.sample tracker.conf
# vim tracker.conf
```

② 编辑tracker.conf，标红的需要修改下，其它的默认即可。

```
# 配置文件是否不生效，false 为生效
disabled=false

# 提供服务的端口
port=22122

# Tracker 数据和日志目录地址（根目录必须存在，子目录会自动创建）
base_path=/ljzsg/fastdfs/tracker
```

```
# HTTP 服务端口
http.server_port=80
```

③ 创建tracker基础数据目录，即base_path对应的目录

```
# mkdir -p /ljzsg/fastdfs/tracker
```

④ 防火墙中打开跟踪端口（默认的22122）

```
# vim /etc/sysconfig/iptables

添加如下端口行:
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22122 -j ACCEPT

重启防火墙:
# service iptables restart
```

⑤ 启动Tracker

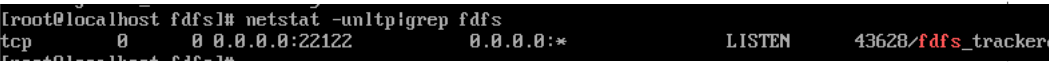
初次成功启动，会在 /ljzsg/fdfsdfs/tracker/ (配置的base_path)下创建 data、logs 两个目录。

```
可以用这种方式启动
# /etc/init.d/fdfs_trackerd start

也可以用这种方式启动，前提是上面创建了软链接，后面都用这种方式
# service fdfs_trackerd start
```

查看 FastDFS Tracker 是否已成功启动，22122端口正在被监听，则算是Tracker服务安装成功。

```
# netstat -unltp|grep fdfs
```



关闭Tracker命令:

```
# service fdfs_trackerd stop
```

⑥ 设置Tracker开机启动

```
# chkconfig fdfs_trackerd on

或者:
# vim /etc/rc.d/rc.local
加入配置:
/etc/init.d/fdfs_trackerd start
```

⑦ tracker server 目录及文件结构

Tracker服务启动成功后，会在base_path下创建data、logs两个目录。目录结构如下:

```
${base_path}
|__data
|  |__storage_groups.dat: 存储分组信息
|  |__storage_servers.dat: 存储服务器列表
|__logs
|  |__trackerd.log: tracker server 日志文件
```

4、配置 FastDFS 存储 (Storage)

① 进入 /etc/fdfs 目录，复制 FastDFS 存储器样例配置文件 storage.conf.sample，并重命名为 storage.conf

```
# cd /etc/fdfs
# cp storage.conf.sample storage.conf
# vim storage.conf
```

② 编辑storage.conf

标红的需要修改，其它的默认即可。

```
# 配置文件是否不生效, false 为生效
disabled=false

# 指定此 storage server 所在 组(卷)
group_name=group1

# storage server 服务端口
port=23000

# 心跳间隔时间, 单位为秒 (这里是指主动向 tracker server 发送心跳)
```

```

heart_beat_interval=30

# Storage 数据和日志目录地址 (根目录必须存在, 子目录会自动生成)
base_path=/ljzsg/fastdfs/storage

# 存放文件时 storage server 支持多个路径。这里配置存放文件的基路径数目, 通常只配一个目录。
store_path_count=1

# 逐一配置 store_path_count 个路径, 索引号基于 0。
# 如果不配置 store_path0, 那它就和 base_path 对应的路径一样。
store_path0=/ljzsg/fastdfs/file

# FastDFS 存储文件时, 采用了两级目录。这里配置存放文件的目录个数。
# 如果本参数只为 N (如: 256), 那么 storage server 在初次运行时, 会在 store_path 下自动创建 N * N 个存放文件的子目录。
subdir_count_per_path=256

# tracker_server 的列表, 会主动连接 tracker_server
# 有多个 tracker server 时, 每个 tracker server 写一行
tracker_server=file.ljzsg.com:22122

# 允许系统同步的时间段 (默认是全天)。一般用于避免高峰同步产生一些问题而设定。
sync_start_time=00:00
sync_end_time=23:59
# 访问端口
http_server_port=80

```

③ 创建Storage基础数据目录, 对应base_path目录

```

# mkdir -p /ljzsg/fastdfs/storage

# 这是配置的store_path0路径
# mkdir -p /ljzsg/fastdfs/file

```

④ 防火墙中打开存储服务器端口 (默认的 23000)

```

# vim /etc/sysconfig/iptables

添加如下端口行:
-A INPUT -m state --state NEW -m tcp -p tcp --dport 23000 -j ACCEPT

重启防火墙:
# service iptables restart

```

```

-A FORWARD -j REJECT --reject-with icmp-host-prohibited
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22122 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 23000 -j ACCEPT

```

⑤ 启动 Storage

启动Storage前确保Tracker是启动的。初次启动成功, 会在 /ljzsg/fastdfs/storage 目录下创建 data、logs 两个目录。

```

可以用这种方式启动
# /etc/init.d/fdfs_storaged start

也可以用这种方式, 后面都用这种
# service fdfs_storaged start

```

查看 Storage 是否成功启动, 23000 端口正在被监听, 就算 Storage 启动成功。

```
# netstat -unlt|grep fdfs
```

```

[root@localhost fdfs]# netstat -unlt|grep fdfs
tcp      0      0 0.0.0.0:22122        0.0.0.0:*          LISTEN   47098/fdfs_tracker
tcp      0      0 0.0.0.0:23000        0.0.0.0:*          LISTEN   48108/fdfs_storage

```

关闭Storage命令:

```
# service fdfs_storaged stop
```

查看Storage和Tracker是否在通信:

```
/usr/bin/fdfs_monitor /etc/fdfs/storage.conf
```



```
[root@file 00]# /usr/bin/fdfs_monitor /etc/fdfs/storage.conf
[2017-10-16 09:37:06] DEBUG - base_path=/ljzsg/fastdfs/storage, connect timeout=30, network timeout=60, tracker_se
nnection_pool=0, g_connection_pool_max_idle_time=3600s, use_storage_id=0, storage server id count: 0

server_count=1, server_index=0

tracker server is 192.168.51.128:22122

group count: 1

Group 1:
group name = group1
disk total space = 18121 MB
disk free space = 13486 MB
trunk free space = 0 MB
storage server count = 1
active server count = 1
storage server port = 23000
storage HTTP port = 80
store path count = 1
subdir count per path = 256
current write server index = 0
current trunk file id = 0

Storage 1:
id = 192.168.51.128
ip_addr = 192.168.51.128 (file.ljzsg.com) ACTIVE
http_domain =
version = 5.05
join time = 2017-10-11 13:35:37
up time = 2017-10-16 09:17:53
total storage = 18121 MB
```

⑥ 设置 Storage 开机启动

```
# chkconfig fdfs_storaged on

或者:
# vim /etc/rc.d/rc.local
加入配置:
/etc/init.d/fdfs_storaged start
```

⑦ Storage 目录

同 Tracker，Storage 启动成功后，在base_path 下创建了data、logs目录，记录着 Storage Server 的信息。

在 store_path0 目录下，创建了N*N个子目录：

```
[root@localhost file]# ls /ljzsg/fastdfs/file/data/
00 07 0E 15 1C 23 2A 31 38 3F 46 4D 54 5B 62 69 70 77 7E 85 8C 93 9A A1 A8 AF B6 BD C4 CB D2 D9 E0 E9
01 08 0F 16 1D 24 2B 32 39 40 47 4E 55 5C 63 6A 71 78 7F 86 8D 94 9B A2 A9 B0 B7 BE C5 CC D3 DA E1 E2
02 09 10 17 1E 25 2C 33 3A 41 48 4F 56 5D 64 6B 72 79 80 87 8E 95 9C A3 AA B1 B8 BF C6 CD D4 DB E3 E4
03 0A 11 18 1F 26 2D 34 3B 42 49 50 57 5E 65 6C 73 7A 81 88 8F 96 9D A4 AB B2 B9 C8 C7 CE D5 DC E5 E6
04 0B 12 19 20 27 2E 35 3C 43 4A 51 58 5F 66 6D 74 7B 82 89 90 97 9E A5 AC B3 BA C1 C8 CF D6 DD E4 E5
05 0C 13 1A 21 28 2F 36 3D 44 4B 52 59 60 67 6E 75 7C 83 8A 91 98 9F A6 AD B4 BB C2 C9 D0 D7 DE E5 E6
06 0D 14 1B 22 29 30 37 3E 45 4C 53 5A 61 68 6F 76 7D 84 8B 92 99 A0 A7 AE B5 BC C3 CA D1 D8 DF E6
```

5. 文件上传测试

① 修改 Tracker 服务器中的客户端配置文件

```
# cd /etc/fdfs
# cp client.conf.sample client.conf
# vim client.conf
```

修改如下配置即可，其它默认。

```
# Client 的数据和日志目录
base_path=/ljzsg/fastdfs/client

# Tracker端口
tracker_server=file.ljzsg.com:22122
```

② 上传测试

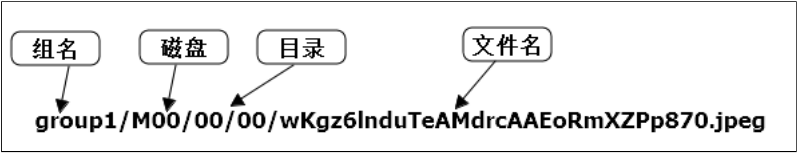
在linux内部执行如下命令上传 namei.jpeg 图片

```
# /usr/bin/fdfs_upload_file /etc/fdfs/client.conf namei.jpeg
```

上传成功后返回文件ID号：group1/M00/00/00/wKgZ6lnduTeAMdrCAAEOmXZPp870.jpeg

```
[root@localhost data]# ls
fastdfs.down namei.jpeg
[root@localhost data]# /usr/bin/fdfs_upload_file /etc/fdfs/client.conf namei.jpeg
group1/M00/00/00/wKgZ6lnduTeAMdrCAAEOmXZPp870.jpeg
```

返回的文件ID由group、存储目录、两级子目录、fileid、文件后缀名（由客户端指定，主要用于区分文件类型）拼接而成。



三、安装Nginx

上面将文件上传成功了，但我们无法下载。因此安装Nginx作为服务器以支持Http方式访问文件。同时，后面安装FastDFS的Nginx模块也需要Nginx环境。

Nginx只需要安装到StorageServer所在的服务器即可，用于访问文件。我这里由于是单机，TrackerServer和StorageServer在一台服务器上。

1、安装nginx所需环境

① gcc 安装

```
# yum install gcc-c++
```

② PCRE pcre-devel 安装

```
# yum install -y pcre pcre-devel
```

③ zlib 安装

```
# yum install -y zlib zlib-devel
```

④ OpenSSL 安装

```
# yum install -y openssl openssl-devel
```

2、安装Nginx

① 下载nginx

```
# wget -c https://nginx.org/download/nginx-1.12.1.tar.gz
```

② 解压

```
# tar -zxvf nginx-1.12.1.tar.gz
# cd nginx-1.12.1
```

③ 使用默认配置

```
# ./configure
```

④ 编译、安装

```
# make
# make install
```

⑤ 启动nginx




```
# cd /usr/local/nginx/sbin/
# ./nginx
```

其它命令

```
# ./nginx -s stop
# ./nginx -s quit
# ./nginx -s reload
```



⑥ 设置开机启动




```
# vim /etc/rc.local
```

添加一行:

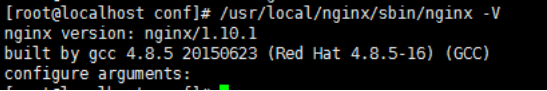
```
/usr/local/nginx/sbin/nginx
```

```
# 设置执行权限
# chmod 755 rc.local
```



⑦ 查看nginx的版本及模块

```
/usr/local/nginx/sbin/nginx -V
```



⑧ 防火墙中打开Nginx端口（默认的 80）

添加后就能在本机使用80端口访问了。




```
# vim /etc/sysconfig/iptables

添加如下端口行:
-A INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT

重启防火墙:
# service iptables restart
```

```
# Generated by iptables-save v1.4.21 on Wed Oct 11 12:01:40 2017
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [1:84]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 80 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22122 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 23000 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
# Completed on Wed Oct 11 12:01:40 2017
~
```

3、访问文件

简单的测试访问文件

① 修改nginx.conf

```
# vim /usr/local/nginx/conf/nginx.conf

添加如下行, 将 /group1/M00 映射到 /ljzsg/fastdfs/file/data
location /group1/M00 {
    alias /ljzsg/fastdfs/file/data;
}

# 重启nginx
# /usr/local/nginx/sbin/nginx -s reload
```

```
server {
    listen      80;
    server_name localhost;

    #charset koi8-r;

    #access_log logs/host.access.log main;

    location / {
        root    html;
        index   index.html index.htm;
    }

    #error_page 404              /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
        root    html;
    }

    location /group1/M00 {
        alias /ljzsg/fastdfs/file/data;
    }
}
```

② 在浏览器访问之前上传的图片、成功。
http://file.ljzsg.com/group1/M00/00/00/wKgZ6lnduTeAMdrcAAEoRmXZPp870.jpeg

[回到顶部](#)

四、FastDFS 配置 Nginx 模块

1、安装配置Nginx模块

① fastdfs-nginx-module 模块说明

FastDFS 通过 Tracker 服务器, 将文件放在 Storage 服务器存储, 但是同组存储服务器之间需要进行文件复制, 有同步延迟的问题。
假设 Tracker 服务器将文件上传到了 192.168.51.128, 上传成功后文件 ID已经返回给客户端。

此时 FastDFS 存储集群机制会将这个文件同步到同组存储 192.168.51.129, 在文件还没有复制完成的情况下, 客户端如果用这个文件 ID 在 192.168.51.129 上取文件, 就会出现文件无法访问的错误。

而 fastdfs-nginx-module 可以重定向文件链接到源服务器取文件, 避免客户端由于复制延迟导致的文件无法访问错误。

② 下载 fastdfs-nginx-module、解压

```
# 这里为啥这么长一串呢, 因为最新版的master与当前nginx有些版本问题。
# wget https://github.com/happyfish100/fastdfs-nginx-
module/archive/5e5f3566bbfa57418b5506aaefbe107a42c9fcb1.zip

# 解压
# unzip 5e5f3566bbfa57418b5506aaefbe107a42c9fcb1.zip

# 重命名
# mv fastdfs-nginx-module-5e5f3566bbfa57418b5506aaefbe107a42c9fcb1 fastdfs-nginx-module-master
```

③ 配置Nginx

在nginx中添加模块

```
# 先停掉nginx服务
# /usr/local/nginx/sbin/nginx -s stop

进入解压包目录
# cd /softpackages/nginx-1.12.1/

# 添加模块
# ./configure --add-module=../fastdfs-nginx-module-master/src

重新编译、安装
# make && make install
```

④ 查看Nginx的模块

```
# /usr/local/nginx/sbin/nginx -V
```

有下面这个就说明添加模块成功

```
[root@localhost nginx-1.12.1]# /usr/local/nginx/sbin/nginx -V
nginx version: nginx/1.12.1
built by gcc 4.8.5 20150623 (Red Hat 4.8.5-16) (GCC)
configure arguments: --add-module=../fastdfs-nginx-module-master/src
```

⑤ 复制 fastdfs-nginx-module 源码中的配置文件到/etc/fdfs 目录, 并修改

```
# cd /softpackages/fastdfs-nginx-module-master/src

# cp mod_fastdfs.conf /etc/fdfs/
```

修改如下配置, 其它默认

```
# 连接超时时间
connect_timeout=10

# Tracker Server
tracker_server=file.ljzsg.com:22122

# StorageServer 默认端口
storage_server_port=23000

# 如果文件ID的uri中包含/group**, 则要设置为true
url_have_group_name = true

# Storage 配置的store_path0路径, 必须和storage.conf中的一致
store_path0=/ljzsg/fastdfs/file
```

⑥ 复制 FastDFS 的部分配置文件到/etc/fdfs 目录

```
# cd /softpackages/fastdfs-5.05/conf/

# cp anti-steal.jpg http.conf mime.types /etc/fdfs/
```

⑦ 配置nginx, 修改nginx.conf

```
# vim /usr/local/nginx/conf/nginx.conf
```

修改配置，其它的默认

在80端口下添加fastdfs-nginx模块

```
location ~/group([0-9])/M00 {
    ngx_fastdfs_module;
}
```

```
server {
    listen      80;
    server_name localhost;

    #charset koi8-r;

    #access_log logs/host.access.log main;

    location / {
        root    html;
        index   index.html index.htm;
    }

    #error_page 404          /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root    html;
    }

    #location /group1/M00 {
    #    alias /ljzsg/fastdfs/file/data;
    #}

    location ~/group([0-9])/M00 {
        ngx_fastdfs_module;
    }
```

注意:

listen 80 端口值是要与 /etc/fdfs/storage.conf 中的 http.server_port=80 (前面改成80了)相对应。如果改成其它端口，则需要统一，同时在防火墙中打开该端口。

location 的配置，如果有多个group则配置location ~/group([0-9])/M00，没有则不用配group。

⑧ 在/ljzsg/fastdfs/file 文件存储目录下创建软连接，将其链接到实际存放数据的目录，这一步可以省略。

```
# ln -s /ljzsg/fastdfs/file/data/ /ljzsg/fastdfs/file/data/M00
```

⑨ 启动nginx

```
# /usr/local/nginx/sbin/nginx
```

打印处如下就算配置成功

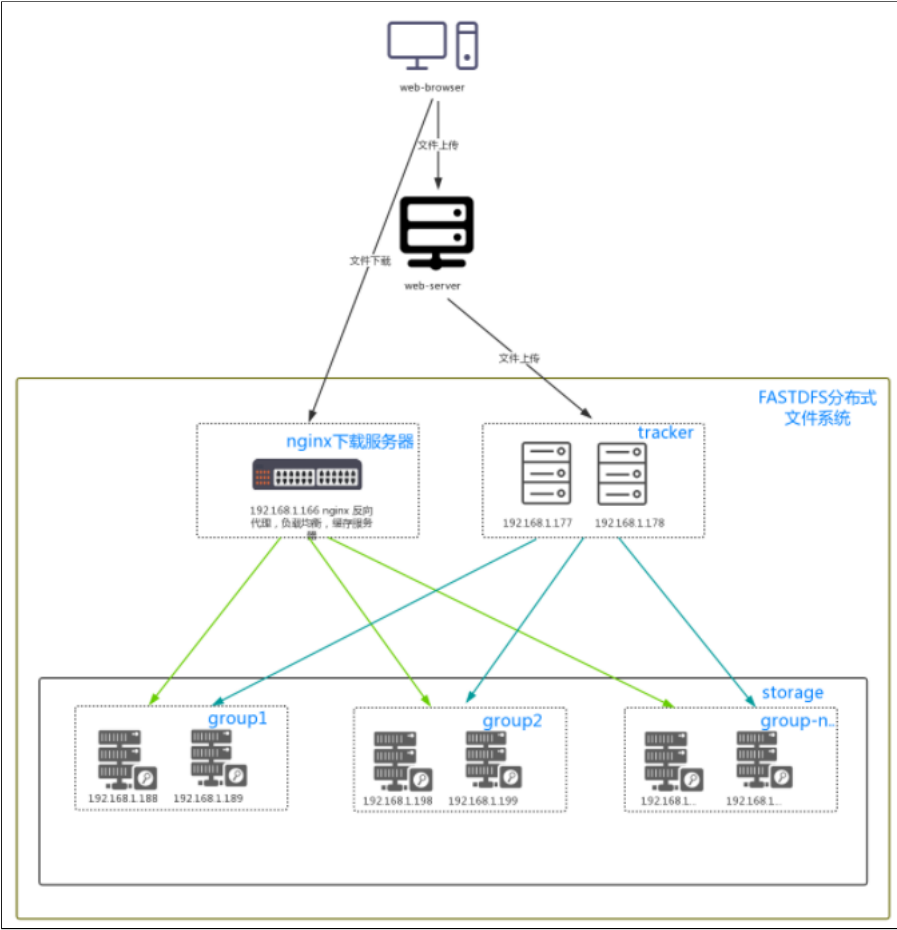
```
nginx: invalid option: start
[root@localhost fdfs]# /usr/local/nginx/sbin/nginx
ngx_http_fastdfs_set pid=7273
```

⑩ 在地址栏访问。

能下载文件就算安装成功。注意和第三点中直接使用nginx路由访问不同的是，这里配置 fastdfs-nginx-module 模块，可以重定向文件链接到源服务器取文件。

<http://file.ljzsg.com/group1/M00/00/00/wKgZ6InduTeAMdrcAAEoRmXZPp870.jpeg>

最终部署结构图(盗的图): 可以按照下面的结构搭建环境。



[回到顶部](#)

五、Java客户端

前面文件系统平台搭建好了，现在就要写客户端代码在系统中实现上传下载，这里只是简单的测试代码。

1、首先需要搭建 FastDFS 客户端Java开发环境

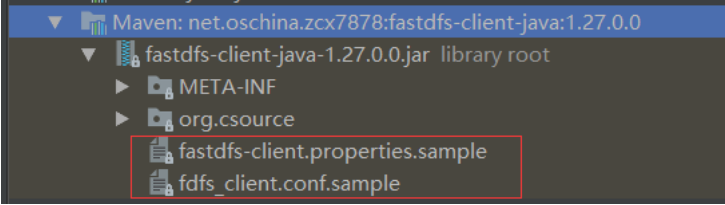
① 项目中使用maven进行依赖管理，可以在pom.xml中引入如下依赖即可：

```
<dependency>
  <groupId>net.oschina.zcx7878</groupId>
  <artifactId>fastdfs-client-java</artifactId>
  <version>1.27.0.0</version>
</dependency>
```

其它的方式，参考官方文档：<https://github.com/happyfish100/fastdfs-client-java>

② 引入配置文件

可直接复制包下的 fastdfs-client.properties.sample 或者 fdfs_client.conf.sample，到你的项目中，去掉.sample。



我这里直接复制 fastdfs-client.properties.sample 中的配置到项目配置文件 config.properties 中，修改tracker_servers。只需要加载这个配置文件即可

```
config.properties
80
81  ## fastdfs-client
82
83  fastdfs.connect_timeout_in_seconds=5
84  fastdfs.network_timeout_in_seconds=30
85
86  fastdfs.charset=UTF-8
87
88  fastdfs.http_anti_steal_token=false
89  fastdfs.http_secret_key=FastDFS1234567890
90  fastdfs.http_tracker_http_port=80
91
92  #fastdfs.tracker_servers=10.0.11.201:22122,10.0.11.202:22122,10.0.11.203:22122
93  fastdfs.tracker_servers=file.ljzsg.com:22122
94
```

2、客户端API

个人封装的FastDFS Java API以同步到github: <https://github.com/bojiangzhou/lyyzoo-fastdfs-java.git>

六、权限控制

前面使用nginx支持http方式访问文件，但所有人都能直接访问这个文件服务器了，所以做一下权限控制。

FastDFS的权限控制是在服务端开启token验证，客户端根据文件名、当前unix时间戳、秘钥获取token，在地址中带上token参数即可通过http方式访问文件。

① 服务端开启token验证

```
修改http.conf
# vim /etc/fdfs/http.conf

设置为true表示开启token验证
http.anti_steal.check_token=true

设置token失效的时间单位为秒(s)
http.anti_steal.token_ttl=1800

密钥，跟客户端配置文件的fastdfs.http_secret_key保持一致
http.anti_steal.secret_key=FASTDFS1234567890

如果token检查失败，返回的页面
http.anti_steal.token_check_fail=/ljzsg/fastdfs/page/403.html
```

记得重启服务。

② 配置客户端

客户端只需要设置如下两个参数即可，两边的密钥保持一致。

```
# token 防盗链功能
fastdfs.http_anti_steal_token=true
# 密钥
fastdfs.http_secret_key=FASTDFS1234567890
```

③ 客户端生成token

访问文件需要带上生成的token以及unix时间戳，所以返回的token是token和时间戳的拼接。

之后，将token拼接在地址后即可访问: file.ljzsg.com/group1/M00/00/00/wKgZgFnkXqAlfXyAAEoRmXZPp878.jpeg?token=078d370098b03e9020b82c829c205e1f&ts=1508141521

```
1  /**
2   * 获取访问服务器的token，拼接到地址后面
3   *
4   * @param filepath 文件路径 group1/M00/00/00/wKgZgFnkTPyAIAUGAAEoRmXZPp876.jpeg
5   * @param httpSecretKey 密钥
6   * @return 返回token，如: token=078d370098b03e9020b82c829c205e1f&ts=1508141521
7   */
8  public static String getToken(String filepath, String httpSecretKey){
9      // unix seconds
10     int ts = (int) Instant.now().getEpochSecond();
11     // token
12     String token = "null";
13     try {
14         token = ProtoCommon.getToken(getFilename(filepath), ts, httpSecretKey);
15     } catch (UnsupportedEncodingException e) {
16         e.printStackTrace();
17     }
```

```
17         } catch (NoSuchAlgorithmException e) {
18             e.printStackTrace();
19         } catch (MyException e) {
20             e.printStackTrace();
21         }
22
23         StringBuilder sb = new StringBuilder();
24         sb.append("token=").append(token);
25         sb.append("&ts=").append(ts);
26
27         return sb.toString();
28     }
```

④ 注意事项

如果生成的token验证无法通过，请进行如下两项检查：

A. 确认调用token生成函数(ProtoCommon.getToken)，传递的文件ID中没有包含group name。传递的文件ID格式形如：M00/00/00/wKgzgFnkTPyAIAUGAAEoRmXZPp876.jpeg

B. 确认服务器时间基本是一致的，注意服务器时间不能相差太多，不要相差到分钟级别。

⑤ 对比下发现，如果系统文件隐私性较高，可以直接通过fastdfs-client提供的API去访问即可，不用再配置Nginx走http访问。配置Nginx的主要目的是为了快速访问服务器的文件(如图片)，如果还要加权限验证，则需要客户端生成token，其实已经没有多大意义。

关键是，这里我没找到FastDFS如何对部分资源加token验证，部分开放。有知道的还请留言。

OK，以上就是单机中使用FastDFS搭建文件系统并上传下载的过程。

完！！

作者: [bojiangzhou](#)
出处: <http://www.cnblogs.com/chiangchou/>
本文版权归作者和博客园共有，欢迎转载，但未经作者同意必须保留此段声明，且在文章页面明显位置给出原文连接，否则保留追究法律责任的权利。

分类: [fastdfs](#)
标签: [文件系统](#), [fastdfs](#), [fastdfs-java](#)

好文要顶

关注我

收藏该文

[bojiangzhou](#)
关注 - 7
粉丝 - 195
[+加关注](#)

90

« 上一篇: [在IntelliJ IDEA中使用Debug](#)
» 下一篇: [Redis 学习\(一\) —— 安装、通用key操作命令](#)

posted on 2017-10-18 23:24 [bojiangzhou](#) 阅读(30811) 评论(16) [编辑](#) [收藏](#)

发表评论

#1楼 2017-10-19 08:45 | 幻天芒

小文件是如何存储的呢？是否有实现多个小文件合并为一个大文件这样的操作？

支持(0) 反对(0)

#2楼[楼主] 2017-10-19 11:00 | bojiangzhou

@ 幻天芒
暂时没有 后面继续

支持(0) 反对(0)

#3楼 2017-10-19 11:23 | 幻天芒

@ bojiangzhou
[引用](#)
@幻天芒
暂时没有 后面继续

要应对海量小文件，这个功能相对比较重要。

支持(0) 反对(0)

#4楼 2017-10-19 14:21 | 花儿笑弯了腰

mark, 请问下有图片的文件存储的推荐吗?

支持(0) 反对(0)

#5楼[楼主] 2017-10-19 20:15 | bojiangzhou

@ 花儿笑弯了腰
那就是FastDFS了, 就是为这个而生的, FastDFS 配合Nginx使用

支持(0) 反对(0)

#6楼 2017-10-30 17:27 | 江湖小生2016

666666

支持(0) 反对(0)

#7楼 2018-01-27 18:12 | 吴川华仔博客

厉害了

支持(0) 反对(0)

#8楼 2018-02-11 10:46 | 亲爱的Joe

请问下怎么怎么从别的主机下载虚拟机上传的图片? 还有怎么从别的主机上传图片到虚拟机上呢?

支持(0) 反对(0)

#9楼 2018-03-16 17:34 | 亲爱的Joe

吐血了, 跟你一步一步做完了, 结果你告诉我单机的。

支持(0) 反对(0)

#10楼[楼主] 2018-03-18 17:18 | bojiangzhou

@ 亲爱的Joe
你完全可以搭建分布式环境, 每个server都可以独立部署, 只是我这里在一个服务器中完成的。配置对了就行。

支持(0) 反对(0)

#11楼 2018-04-08 21:04 | 四月、

唯一——一个跟着一步一步还不错的博客

支持(1) 反对(0)

#12楼 2018-04-10 20:13 | Ethan Shan

Mark

支持(0) 反对(0)

#13楼 2018-05-06 14:57 | 带你走天涯

mark, 不错哈

支持(0) 反对(0)

#14楼 2018-05-17 10:36 | 很急

博主, 两个问题想请教一下
1. 我有多个group, 多个storage节点, 需要在所有节点上装nginx和模块 还是 任意一台服务器装就行呢? 我看博客里"五、Java客户端"这一节上面的图里, 画的只是166这台机器上装了, 就可以下载访问所有的storage了。
2. 现在我只在台机器上装了nginx和fastdfs-nginx-module, 然后我有一个地址如: group1/M00/00/00/xxx.tar.gz, 通过地址栏可以访问到文件, 现在我把这个地址的group1改为group2, 再到地址栏访问, 依然能访问到文件, 博主, 我有点困惑: 这种情况是group1和group2的文件同步了, 还是fastdfs-nginx-module把地址给转发了, 实际到group1里去拿文件了呢?

支持(0) 反对(0)

#15楼[楼主] 2018-05-21 10:35 | bojiangzhou

@ 很急
Storage Server: 存储服务器, 主要提供容量和备份服务; 以 group 为单位, 每个 group 内可以有 multiple storage server, 数据互为备份。

支持(0) 反对(0)

#16楼 2018-05-30 09:03 | color4joker

有个问题: 为什么要做文件的软连接呢? 我发现上传图片后, 图片就data/00/00/路径下, 为什么要在data下建立group1/M00/, 不建立的话直接用<http://localhost/00/00/文件名> 就可以查看了啊, 这个没有懂。。。另外不知道为什么, 我如果做了软连接, 就会出问题, 就访问不了图片了, 会出现404nginx错误

支持(0) 反对(0)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论, 请 [登录](#) 或 [注册](#), [访问网站首页](#)。

- 【推荐】超50万VC++源码: 大型组态工控、电力仿真CAD与GIS源码库!
- 【推荐】如何快速搭建人工智能应用?
- 【活动】AI技术全面场景化落地实践
- 【大赛】2018首届"顶天立地"AI开发者大赛



最新IT新闻:

- 微软发布PowerShell快照预览 支持Ubuntu等GNU/Linux发行版
 - Google Ventures利用算法帮助判断是否进行投资
 - 语音UI是未来？罗永浩其实没说错，但关键是时间不对
 - OpenID失去了最重要的拥护者StackOverflow
 - 《魔兽世界》8.0版国服正式过审 离上线更近了
- » 更多新闻...



最新知识库文章:

- 观察之道：带你走进可观察性
 - 危害程序员职业生涯的三大观念
 - 断点单步跟踪是一种低效的调试方法
 - 测试 | 让每一粒尘埃有的放矢
 - 从Excel到微服务
- » 更多知识库文章...