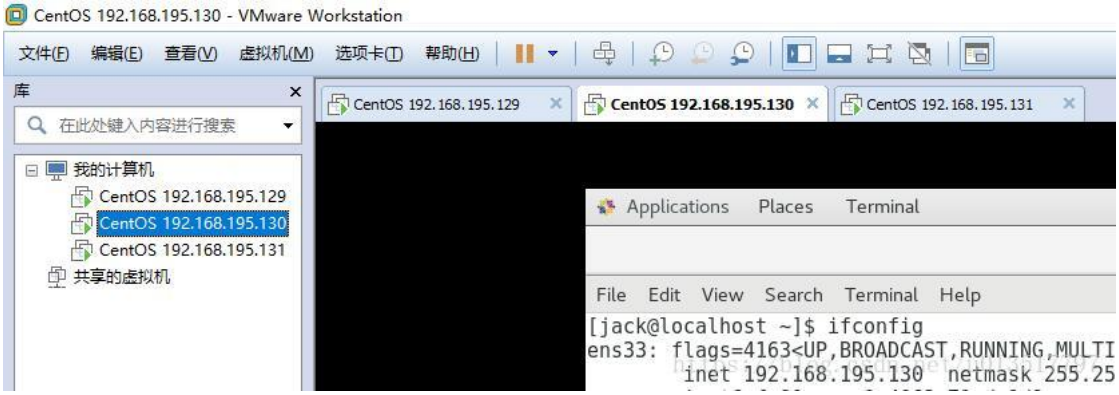
【FastDFS分布式文件系统】5.FastDFS客户端的配置、启动和常用命令

2018年07月08日 14:54:17 光仔December 阅读数：150 标签： fastdfs fdfs_upload_file fdfs_download_file fdfs_file_info fdfs_delete_file

个人分类： FastDFS

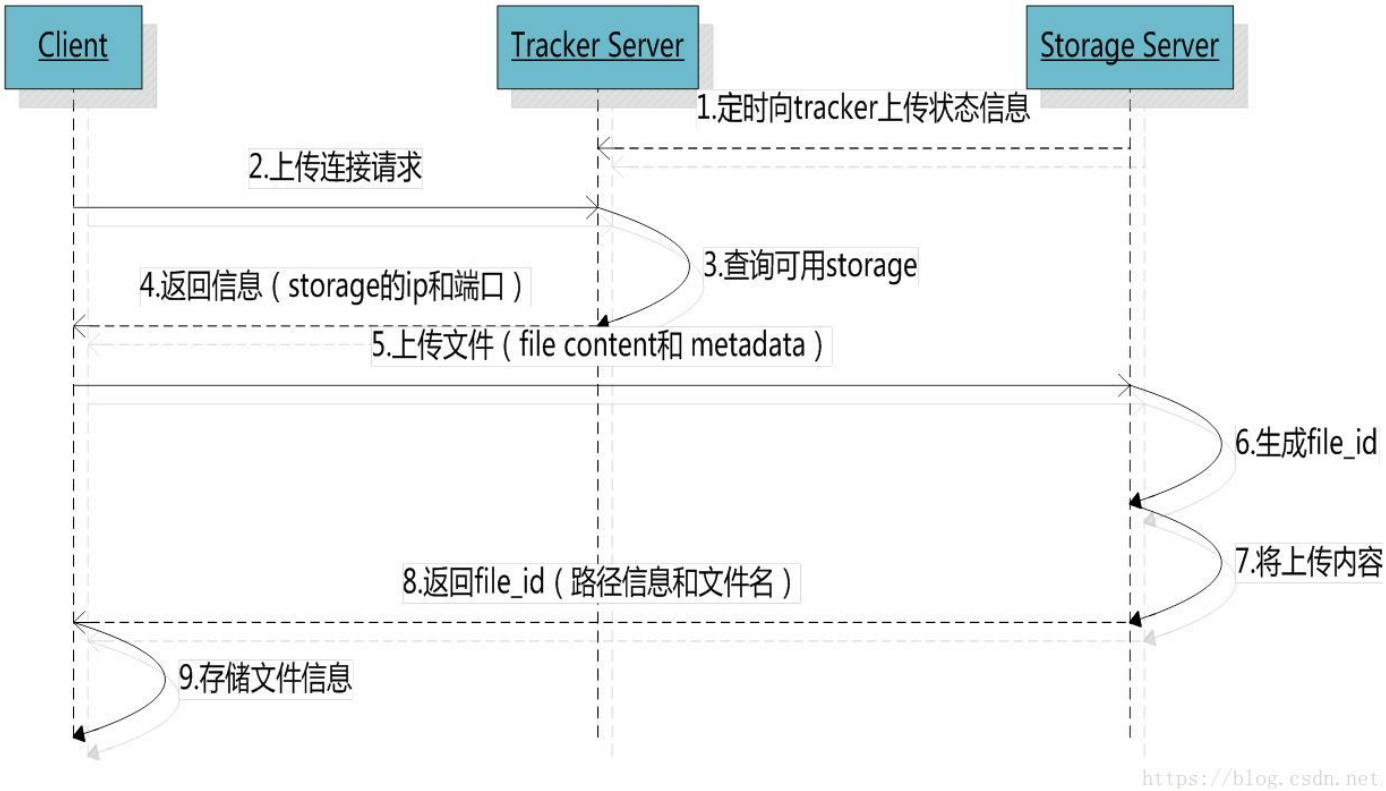
上一篇我们介绍了FastDFS服务端的tracker追踪服务器和storage存储服务器，本篇来介绍一下客户端的启动，以及外部客户端如何与FastDFS服务端进行交

和之前一样，服务端部署在三台服务器上：



其中192.168.195.129是tracker追踪服务器，192.168.195.130和192.168.195.131是storage存储服务器。

回顾一下之前的客户端与FastDFS之间的交互图：



客户端其实与FastDFS服务端交互，就是与追踪服务器交互，追踪服务器会根据情况返回可用的存储服务器的信息，此时客户端才与存储服务器进行交互。

下面我们在192.168.195.129上配置一个客户端，先打开之前的fdfs的配置文件：

```
[root@localhost jack]# cd /etc/fdfs/
[root@localhost fdfs]# ll
total 32
-rw-r--r--. 1 root root 1461 Jun 30 00:25 client.conf.sample
-rw-r--r--. 1 root root 7927 Jun 30 00:25 storage.conf.sample
-rw-r--r--. 1 root root  105 Jun 30 00:25 storage_ids.conf.sample
-rw-r--r--. 1 root root 7387 Jun 30 01:56 tracker.conf
-rw-r--r--. 1 root root 7389 Jun 30 00:25 tracker.conf.sample
```

其中client.conf.sample就是客户端配置文件的模板文件，无论我们使用c、java还是php作为编写客户端的语言，我们都需要配置该文件。这里我们复制该模板文件并修改名为f"：

cp client.conf.sample client.conf
并进行以下配置：

```
# 连接超时时间，默认30秒
connect_timeout=30

# 网络超时时间，默认60秒
network_timeout=60

# 存放日志的根目录
base_path=/tmp

# 客户端对应的追踪服务器的ip地址:端口，如果有多个就写多个
tracker_server=192.168.195.129:22122

#日志级别:
### emerg for emergency
### alert
### crit for critical
### error
### warn for warning
### notice
### info
### debug
log_level=info

# 是否使用连接池
use_connection_pool = false

# connections whose the idle time exceeds this time will be closed
# unit: second
# default value is 3600
# since V4.05
connection_pool_max_idle_time = 3600

# if load FastDFS parameters from tracker server
# since V4.05
# default value is false
load_fdfs_parameters_from_tracker=false

# if use storage ID instead of IP address
# same as tracker.conf
# valid only when load_fdfs_parameters_from_tracker is false
# default value is false
# since V4.05
use_storage_id = false

# specify storage ids filename, can use relative or absolute path
# same as tracker.conf
# valid only when load_fdfs_parameters_from_tracker is false
# since V4.05
storage_ids_filename = storage_ids.conf

#HTTP settings
http.tracker_server_port=80

#use "#include" directive to include HTTP other settiongs
##include http.conf
```

主要就是配置了日志文件根目录和追踪服务器的地址。

然后我们就可以测试文件的上传和下载。安装好FastDFS会给我们提供很多命令，我们输入"fdfs_"按下Tab键，会看到所有的命令提示：

```
[root@localhost fdfs]# fdfs_
fdfs_appender_test     fdfs_download_file     fdfs_test1
fdfs_appender_test1    fdfs_file_info         fdfs_trackerd
fdfs_append_file       fdfs_monitor           fdfs_upload_appender
fdfs_crc32             fdfs_storaged          fdfs_upload_file
fdfs_delete_file       fdfs_test
```


下面我们来介绍一下比较常用的客户端命令的意义和使用。

(1)fdfs_upload_file

这里的“fdfs_upload_file”就是上传文件的一个命令，我们输入该命令，不输入任何参数，会给我们提示该命令的用法：

```
[root@localhost fdfs]# fdfs_upload_file
Usage: fdfs_upload_file <config_file> <local_filename> [storage_ip:port] [store_path_in
dex]
```

“fdfs_upload_file”命令后面首先跟着的是客户端配置文件的路径，然后是要上传的本地文件，按下回车的话会默认上传到配置文件配置的追踪服务器返回给客户端的存储服务

```
[root@localhost fdfs]# fdfs_upload_file /etc/fdfs/client.conf /etc/fdfs/test.txt
group1/M00/00/00/wKjDgltBh7iAWKw6AAAAC4arQWs217.txt
[root@localhost fdfs]#
```

可以看到，这里我们上传了一个测试文件，里面写了一句“HelloWorld”，此时按下回车，FastDFS会给我们返回一个文件的路径，该路径就是刚刚上传的文件在存储服务器中

因为我们打开一台存储服务器(192.168.195.130)的存储路径：

```
[root@localhost 00]# cd /data/fdfs_storage/store
[root@localhost store]# ll
total 12
drwxr-xr-x. 258 root root 8192 Jun 30 05:36 data
[root@localhost store]# cd data
[root@localhost data]# ls
00  0D  1A  27  34  41  4E  5B  68  75  82  8F  9C  A9  B6  C3  D0  DD  EA  F7
01  0E  1B  28  35  42  4F  5C  69  76  83  90  9D  AA  B7  C4  D1  DE  EB  F8
02  0F  1C  29  36  43  50  5D  6A  77  84  91  9E  AB  B8  C5  D2  DF  EC  F9
03  10  1D  2A  37  44  51  5E  6B  78  85  92  9F  AC  B9  C6  D3  E0  ED  FA
04  11  1E  2B  38  45  52  5F  6C  79  86  93  A0  AD  BA  C7  D4  E1  EE  FB
05  12  1F  2C  39  46  53  60  6D  7A  87  94  A1  AE  BB  C8  D5  E2  EF  FC
06  13  20  2D  3A  47  54  61  6E  7B  88  95  A2  AF  BC  C9  D6  E3  F0  FD
07  14  21  2E  3B  48  55  62  6F  7C  89  96  A3  B0  BD  CA  D7  E4  F1  FE
08  15  22  2F  3C  49  56  63  70  7D  8A  97  A4  B1  BE  CB  D8  E5  F2  FF
09  16  23  30  3D  4A  57  64  71  7E  8B  98  A5  B2  BF  CC  D9  E6  F3
0A  17  24  31  3E  4B  58  65  72  7F  8C  99  A6  B3  C0  CD  DA  E7  F4
0B  18  25  32  3F  4C  59  66  73  80  8D  9A  A7  B4  C1  CE  DB  E8  F5
0C  19  26  33  40  4D  5A  67  74  81  8E  9B  A8  B5  C2  CF  DC  E9  F6
```

可以看到里面有256个文件夹，而每个子目录下还有256个目录(打开00文件夹)：

```
[root@localhost data]# cd 00
[root@localhost 00]# ls
00  0D  1A  27  34  41  4E  5B  68  75  82  8F  9C  A9  B6  C3  D0  DD  EA  F7
01  0E  1B  28  35  42  4F  5C  69  76  83  90  9D  AA  B7  C4  D1  DE  EB  F8
02  0F  1C  29  36  43  50  5D  6A  77  84  91  9E  AB  B8  C5  D2  DF  EC  F9
03  10  1D  2A  37  44  51  5E  6B  78  85  92  9F  AC  B9  C6  D3  E0  ED  FA
04  11  1E  2B  38  45  52  5F  6C  79  86  93  A0  AD  BA  C7  D4  E1  EE  FB
05  12  1F  2C  39  46  53  60  6D  7A  87  94  A1  AE  BB  C8  D5  E2  EF  FC
06  13  20  2D  3A  47  54  61  6E  7B  88  95  A2  AF  BC  C9  D6  E3  F0  FD
07  14  21  2E  3B  48  55  62  6F  7C  89  96  A3  B0  BD  CA  D7  E4  F1  FE
08  15  22  2F  3C  49  56  63  70  7D  8A  97  A4  B1  BE  CB  D8  E5  F2  FF
09  16  23  30  3D  4A  57  64  71  7E  8B  98  A5  B2  BF  CC  D9  E6  F3
0A  17  24  31  3E  4B  58  65  72  7F  8C  99  A6  B3  C0  CD  DA  E7  F4
0B  18  25  32  3F  4C  59  66  73  80  8D  9A  A7  B4  C1  CE  DB  E8  F5
0C  19  26  33  40  4D  5A  67  74  81  8E  9B  A8  B5  C2  CF  DC  E9  F6
```

这是在之前的存储服务器的配置文件中配置的：

subdir_count_per_path=256

然后我们去00/00下去看一下，刚好我们的文件就被上传在该存储服务器上：

```
[root@localhost 00]# cd 00
[root@localhost 00]# ls
wKjDgltBh7iAWKw6AAAAC4arQWs217.txt
```

FastDFS会将上传的文件名进行修改，一个是放置上传的文件重名，以保证文件名的唯一性，另一个是该文件名中是讲过一定算法生成的信息戳，该信息戳中包含了许多信

原存储节点的ip是什么，group中的哪一个server。我们cat查看一下文件内容：

```
[root@localhost 00]# cd 00
[root@localhost 00]# ls
wKjDgltBh7iAWKw6AAAAC4arQWs217.txt
[root@localhost 00]# cat wKjDgltBh7iAWKw6AAAAC4arQWs217.txt
HelloWorld
```

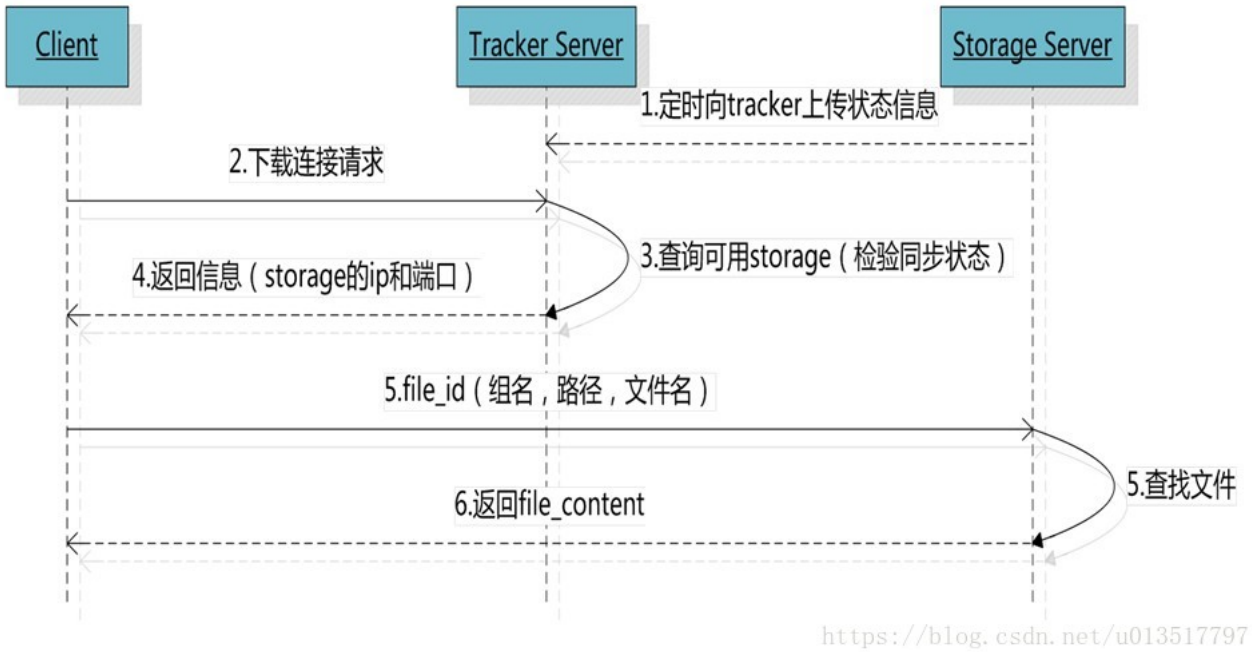
确实是我们之前上传的内容有HelloWorld的文件。

(2)fdfs_download_file

使用"fdfs_download_file"可以进行文件的下载，也是跟上客户端的配置文件以及要下载的文件的路径(该路径是上传的时候存储服务器返回给我们的)：

```
[root@localhost fdfs]# fdfs_download_file /etc/fdfs/client.conf group1/M00/00/00/wKjDgl
tBh7iAWKw6AAAAC4arQWs217.txt
[root@localhost fdfs]# ll
total 44
-rw-r--r--. 1 root root 1447 Jul  7 20:28 client.conf
-rw-r--r--. 1 root root 1461 Jun 30 00:25 client.conf.sample
-rw-r--r--. 1 root root 7927 Jun 30 00:25 storage.conf.sample
-rw-r--r--. 1 root root  105 Jun 30 00:25 storage_ids.conf.sample
-rw-r--r--. 1 root root   11 Jul  7 20:38 test.txt
-rw-r--r--. 1 root root 7387 Jun 30 01:56 tracker.conf
-rw-r--r--. 1 root root 7389 Jun 30 00:25 tracker.conf.sample
-rw-r--r--. 1 root root   11 Jul  7 20:53 wKjDgltBh7iAWKw6AAAAC4arQWs217.txt
```

可以看到，文件被下载到当前目录下了。下载的机制图如下：



步骤1，存储服务器在定时的向追踪服务器上传存储状态。
步骤2，客户端设置追踪服务器的地址，并向追踪服务器发送下载的连接请求。
步骤3，追踪服务器进行一个调度，来查询当前可用的存储。
步骤4，追踪服务器向客户端返回一个可用的存储服务器组的ip和端口。
步骤5，客户端获取到存储服务器组的ip和端口后，向存储服务器传输想要下载的文件的组名、路径、文件名信息。
步骤6，存储服务器接收到文件下载的请求后，会根据文件的组名、路径、文件名信息进行查找。
步骤7，存储服务器在对应路径下找到目标文件后，将目标文件信息返回给客户端。

(3)fdfs_file_info

使用“fdfs_file_info”可以查看到文件的详细存储信息，也是跟上客户端的配置文件以及储服务器返回给我们的文件的路径：

```
[root@localhost fdfs]# fdfs_file_info /etc/fdfs/client.conf group1/M00/00/00/wKjDgltBh7
iAWKw6AAAAC4arQWs217.txt
source storage id: 0
source ip address: 192.168.195.130
file create timestamp: 2018-07-07 20:40:40
file size: 11
file crc32: 2259370347 (0x86AB416B)
```


这时FastDFS会给我们返回该文件的存储服务器节点id、存储服务器的ip地址、文件上传的时间、文件大小以及crc32的校验信息(保证数据的正确，就不得不采用检错的手段

(4)fdfs_delete_file

使用“fdfs_file_info”可以查看到文件的详细存储信息，也是跟上客户端的配置文件以及储服务器返回给我们的文件的路径：

```
[root@localhost fdfs]# fdfs_delete_file /etc/fdfs/client.conf group1/M00/00/00/wKjDgltB
h7iAWKw6AAAAC4arQWs217.txt
[root@localhost fdfs]#
```


此时文件就会在相应的存储服务器中被删除：

```
[root@localhost 00]# ls
wKjDgltBh7iAWKw6AAAAC4arQWs217.txt
[root@localhost 00]# cat wKjDgltBh7iAWKw6AAAAC4arQWs217.txt
HelloWorld
[root@localhost 00]# ls
[root@localhost 00]#
```

无文件，说明被删除

(5)fdfs_upload_appender/fdfs_append_file

使用“fdfs_upload_appender”表示要上传一个可以追加内容的文件，后面跟上客户端的配置文件以及我们要上传的文件的路径：

```
[root@localhost fdfs]# echo "hello" > test1.txt
[root@localhost fdfs]# echo "world" > test2.txt
[root@localhost fdfs]# fdfs_upload_appender /etc/fdfs/client.conf test1.txt
group1/M00/00/00/wKjDgltBkACEPjBxAAAAAHfj3SA788.txt
```

上面我们创建了两个文件，分别存储一个单词，首先将第一个文件作为可追加文件上传到了存储服务器。
我们看一下上传后的文件属性，可以看到被存储在192.168.195.130服务器上：

```
[root@localhost fdfs]# fdfs_file_info /etc/fdfs/client.conf group1/M00/00/00/wKjDgltBkA
CEPjBxAAAAAHfj3SA788.txt
source storage id: 0
source ip address: 192.168.195.130
file create timestamp: 2018-07-07 21:16:00
file size: 6
file crc32: 2011421984 (0x77E3DD20)
[root@localhost fdfs]#
```

我们去192.168.195.130存储服务器中看一下，发现上传成功：

```
[root@localhost 00]# pwd
/data/fdfs_storage/store/data/00/00
[root@localhost 00]# ls
wKjDgltBkACEPjBxAAAAAHfj3SA788.txt
[root@localhost 00]# cat wKjDgltBkACEPjBxAAAAAHfj3SA788.txt
hello
```


然后使用“fdfs_append_file”表示可以将内容追加到目标文件尾部，后面跟上客户端的配置文件以及要追加的目标文件在存储服务器的路径，以及要追加内容的本地文件路径

```
[root@localhost fdfs]# fdfs_append_file /etc/fdfs/client.conf group1/M00/00/00/wKjDgltB
kACEPjBxAAAAAHfj3SA788.txt test2.txt
[root@localhost fdfs]#
```


然后我们到存储服务器中观察，发现文件还是一个，但是文件内容被成功追加：

```
[root@localhost 00]# ls
wKjDgltBkACEPjBxAAAAAHfj3SA788.txt
[root@localhost 00]# cat wKjDgltBkACEPjBxAAAAAHfj3SA788.txt
hello
world
```


(6)fdfs_monitor

使用“fdfs_monitor”会显示当前所有可连接的Tracker Server状态以及相关的存储组信息，后面跟上客户端的配置文件即可：

```
[root@localhost fdfs]# fdfs_monitor /etc/fdfs/client.conf
[2018-07-07 22:11:35] DEBUG - base_path=/tmp, connect_timeout=30, network_timeout=60, t
racker_server_count=1, anti_steal_token=0, anti_steal_secret_key length=0, use_connecti
on_pool=0, g_connection_pool_max_idle_time=3600s, use_storage_id=0, storage server id c
ount: 0

server_count=1, server_index=0

tracker server is 192.168.195.129:22122   追踪服务器信息

group count: 2   两个组

Group 1:   组1的信息
group name = group1
disk total space = 18121 MB
disk free space = 13046 MB
trunk free space = 0 MB
storage server count = 1
active server count = 1
storage server port = 23000
storage HTTP port = 8888
store path count = 1
subdir count per path = 256
current write server index = 0
current trunk file id = 0
```


Group 1下一有个存储服务器130，可以看到相关的存储大小、创建时间等信息：

```
Group 1:
group name = group1
disk total space = 18121 MB
disk free space = 13046 MB
trunk free space = 0 MB
storage server count = 1
active server count = 1
storage server port = 23000
storage HTTP port = 8888
store path count = 1
subdir count per path = 256
current write server index = 0
current trunk file id = 0

        Storage 1:
                id = 192.168.195.130
                ip_addr = 192.168.195.130  ACTIVE
                http domain =
                version = 5.11
                join time = 2018-06-30 03:00:15
                up time = 2018-07-07 19:54:02
                total storage = 18121 MB
                free storage = 13046 MB
                upload priority = 10
                store_path_count = 1
```


Group 2下一有个存储服务器131：

```
Group 2:
group name = group2
disk total space = 18121 MB
disk free space = 13040 MB
trunk free space = 0 MB
storage server count = 1
active server count = 1
storage server port = 23000
storage HTTP port = 8888
store path count = 1
subdir count per path = 256
current write server index = 0
current trunk file id = 0

        Storage 1:
                id = 192.168.195.131
                ip_addr = 192.168.195.131  ACTIVE
                http domain =
                version = 5.11
                join time = 2018-06-30 05:43:56
                up time = 2018-07-07 20:40:21
                total storage = 18121 MB
                free storage = 13040 MB
                upload priority = 10
                store_path_count = 1
                subdir_count_per_path = 256
```


其中的Active就是代表该存储服务器是存活的、可用的。

我们再观察一下“fdfs_monitor”的参数提示：

```
[root@localhost fdfs]# fdfs_monitor
Usage: fdfs_monitor <config_file> [-h <tracker_server>] [list|delete|set_trunk_server <
group_name> [storage_id]]
```


其中的delete就是可以将某些有问题的服务剔除掉，例如删除Group1组中的130服务，语句如下：

fdfs_monitor /etc/fdfs/client.conf delete group1 192.168.195.130

在删除时，我们要确保该服务已经被停止，不然的话会报服务忙的异常。

下一篇我们来讲解如何使用java客户端连接FastDFS。