FastDFS 分布式搭建

1 Fast	:DFS 分布式搭建	2
1.1 考	安装 FastDFS	3
1.2 N	ginx 安装	3
1.3 酉	已置 tracker	4
1.4 酉	已置 storage group1	5
1.5 酉	已置 storage group2	7
1.6 🕖	则试 FastDFS	9

1 FastDFS 分布式搭建

环境如下:
CentOS6.4_64
FastDFS5.0.1
##tracker_server
192.168.10.11
##storage_server group1
192.168.10.12
192.168.10.13
<pre>##storage_server group2</pre>
192.168.10.14
192.168.10.15

##假设 group1 只有一块硬盘, 挂载到/data/fastdfs/storage
##假设 group2 有两块硬盘, 挂载到/data/fastdfs/storage、/data/fastdfs/storage1



整体架构图如下所示:

1.1 安装 FastDFS

a. 在192.168.10.11-192.168.10.15 上分别安装,地址: http://sourceforge.net/projects/fastdfs/files/

[root@localhost src]# pwd /usr/local/src [root@localhost src]# ll -rw-r--r--. 1 root root 405837 Mar 12 20:00 FastDFS_v5.01.tar.gz [root@localhost src]# tar xf FastDFS_v5.01.tar.gz [root@localhost src]# cd FastDFS [root@localhost FastDFS]# ./make.sh [root@localhost FastDFS]# ./make.sh install

b. 创建 fastdfs 账号,用于管理 FastDFS

[root@localhost~]# useradd fastdfs -M -s /sbin/nologin

1.2 Nginx 安装

a. 在 192.168.10.11-192.168.10.15 上分别安装 Nginx, 先安装依赖软件

[root@localhost src]# yum -y install gcc gcc+ gcc-c++ openssl openssl-devel pcre pcre-devel

b. 添加运行 Nginx 的用户(www)及用户组(www)

[root@localhost src]# useradd www -M -s /sbin/nologin

c. 下载 fastdfs-nginx-module_v1.15.tar.gz, 地址 https://code.google.com/p/fastdfs/downloads/list [root@localhost ~]# cd /usr/local/src/ [root@localhost src]# tar xf fastdfs-nginx-module_v1.15.tar.gz [root@localhost src]# ll drwxrwxr-x. 3 steven steven 4096 Feb 18 2013 fastdfs-nginx-module -rw-r--r-. 1 root root 17264 Mar 12 23:38 fastdfs-nginx-module_v1.15.tar.gz

d. 安装 Nginx(注: 192.168.10.11 不能安装 fastdfs-nginx-module 模块)

[root@localhost src]# tar xf nginx-1.4.4.tar.gz [root@localhost src]# cd nginx-1.4.4 [root@localhost nginx-1.4.4]# ./configure --user=www --group=www --prefix=/usr/local/nginx --add-module=../fastdfs-nginx-module/src [root@localhost nginx-1.4.4]# make [root@localhost nginx-1.4.4]# make install

1.3 配置 tracker

a. 在 192.168.10.11 上, 创建 tracker 数据及日志存放目录

[root@localhost ~]# mkdir -p /data/fastdfs/tracker

b. 修改 FastDFS 配置文件 tracker.conf

[root@localhost ~]# vim /etc/fdfs/tracker.conf base_path=/data/fastdfs/tracker max_connections=1024 ##工作线程数,通常设置为 CPU 数 work_threads=8 store_lookup=0 store_server=0 store_path=0 download_server=0 reserved_storage_space=4G run_by_group=fastdfs run_by_user=fastdfs rotate_error_log=true

c. 修改 Nginx 配置文件

```
[root@localhost ~]# vim /usr/local/nginx/conf/nginx.conf
user www.www;
worker processes 8;
pid /usr/local/nginx/logs/nginx.pid;
worker_rlimit_nofile 51200;
events {
    use
           epoll;
    worker_connections 51200;
}
http {
    include
                   mime.types;
    default_type application/octet-stream;
    log format main '$remote addr - $remote user [$time local] "$request" '
                         '$status $body bytes sent "$http referer" '
                         ""$http_user_agent" "$http_x_forwarded_for"";
    access_log /usr/local/nginx/logs/access.log main;
    upstream server g1{
         server 192.168.10.12:80;
         server 192.168.10.13:80;
```

}

```
upstream server_g2{
    server 192.168.10.14:80;
    server 192.168.10.15:80;
}
server {
                 80;
    listen
    server name localhost;
    location /g1 {
         proxy_redirect
                           off;
         proxy_set_header Host $host;
         proxy_set_header X-Real-IP $remote_addr;
         proxy set header X-Forwarded-For $proxy add x forwarded for;
         proxy_pass
                        http://server_g1;
    }
    location /g2 {
         proxy redirect
                           off:
         proxy_set_header Host $host;
         proxy_set_header X-Real-IP $remote_addr;
         proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
         proxy_pass
                        http://server_g2;
    }
}
```

d. 把 tracker 交给 service 管理,并设置开机自启动

[root@localhost ~]# cp /usr/local/src/FastDFS/init.d/fdfs_trackerd /etc/init.d/ [root@localhost ~]# chkconfig --add fdfs_trackerd [root@localhost ~]# chkconfig fdfs_trackerd on

1.4 配置 storage group1

分别在 192.168.10.12、192.168.10.13 上进行 storage server 配置,下面将进行具体的配置

a. 创建数据存放目录

[root@localhost ~]# mkdir -p /data/fastdfs/storage

b. 修改 FastDFS 配置文件 storage.conf

[root@localhost ~]# vim /etc/fdfs/storage.conf group_name=g1 base_path=/data/fastdfs max_connections=1024 ##工作线程数,通常设置为 CPU 数 work_threads=8 store_path_count=1 store_path0=/data/fastdfs/storage ##tracker_server 的地址 tracker_server=192.168.10.11:22122 ##运行 FastDFS 的用户组 run_by_group=fastdfs ##运行 FastDFS 的用户 run_by_user=fastdfs file_distribute_path_mode=1 rotate_error_log=true

c. 把 nginx 模块配置文件拷贝到"/etc/fdfs"里,并修改

[root@localhost ~]# cp /usr/local/src/fastdfs-nginx-module/src/mod_fastdfs.conf /etc/fdfs/ [root@localhost ~]# vim /etc/fdfs/mod_fastdfs.conf connect_timeout=30 tracker_server=192.168.10.11:22122 group_name=g1 url_have_group_name = true store_path_count=1 store_path0=/data/fastdfs/storage

d. 修改 Nginx 配置文件

[root@localhost~]# vim /usr/local/nginx/conf/nginx.conf				
user www.www;				
worker_processes 8;				
pid /usr/local/nginx/logs/nginx.pid;				
worker_rlimit_nofile 51200;				
events {				
use epoll;				
worker_connections 51200;				
}				
http {				
include mime.types;				
default_type application/octet-stream;				
log_format main '\$remote_addr - \$remote_user [\$time_local] "\$request" '				
'\$status \$body bytes sent "\$http referer" '				

e. 把 storage 交给 service 管理,并设置开机自启动

```
[root@localhost ~]# cp /usr/local/src/FastDFS/init.d/fdfs_storaged /etc/init.d/
[root@localhost ~]# chkconfig --add fdfs_storaged
[root@localhost ~]# chkconfig fdfs_storaged on
[root@localhost ~]# service fdfs_storaged start
```

f. 创建软链接

[root@localhost~]# ln -s /data/fastdfs/storage/data /data/fastdfs/storage/data/M00

1.5 配置 storage group2

分别在 192.168.10.14、192.168.10.15 上进行 storage server 配置,下面将进行具体的配置

a. 创建数据存放目录

```
[root@localhost ~]# mkdir -p /data/fastdfs/storage
[root@localhost ~]# mkdir -p /data/fastdfs/storage1
```

b. 修改 FastDFS 配置文件 storage.conf

```
[root@localhost ~]# vim /etc/fdfs/storage.conf
group_name=g2
base_path=/data/fastdfs
max_connections=1024
##工作线程数,通常设置为 CPU 数
work_threads=8
store_path_count=2
store_path0=/data/fastdfs/storage
store_path1=/data/fastdfs/storage1
```

##tracker_server 的地址
tracker_server=192.168.10.11:22122
##运行 FastDFS 的用户组
run_by_group=fastdfs
##运行 FastDFS 的用户
run_by_user=fastdfs
file_distribute_path_mode=1
rotate_error_log=true

c. 把 nginx 模块配置文件拷贝到"/etc/fdfs"里,并修改

[root@localhost ~]# cp /usr/local/src/fastdfs-nginx-module/src/mod_fastdfs.conf /etc/fdfs/ [root@localhost ~]# vim /etc/fdfs/mod_fastdfs.conf connect_timeout=30 tracker_server=192.168.10.11:22122 group_name=g2 url_have_group_name = true store_path_count=2 store_path0=/data/fastdfs/storage store_path1=/data/fastdfs/storage1

d. 修改 Nginx 配置文件

```
[root@localhost ~]# vim /usr/local/nginx/conf/nginx.conf
user www.www;
worker_processes 8;
pid /usr/local/nginx/logs/nginx.pid;
worker_rlimit_nofile 51200;
events {
    use
           epoll;
    worker_connections 51200;
}
http {
    include
                   mime.types;
    default_type application/octet-stream;
    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
                          '$status $body bytes sent "$http referer" '
                          "$http user agent" "$http x forwarded for";
    access_log /usr/local/nginx/logs/access.log main;
    server {
         listen
                       80;
         server_name localhost;
         location /g2/M00{
                  root /data/fastdfs/storage/data;
```

ngx_fastdfs_module; } location /g2/M01 { root /data/fastdfs/storage1/data; ngx_fastdfs_module; }

e. 把 storage 交给 service 管理,并设置开机自启动

[root@localhost ~]# cp /usr/local/src/FastDFS/init.d/fdfs_storaged /etc/init.d/ [root@localhost ~]# chkconfig --add fdfs_storaged [root@localhost ~]# chkconfig fdfs_storaged on [root@localhost ~]# service fdfs_storaged start

f. 创建软链接

}

[root@localhost~]# ln -s /data/fastdfs/storage/data /data/fastdfs/storage/data/M00 [root@localhost~]# ln -s /data/fastdfs/storage1/data /data/fastdfs/storage1/data/M01

1.6 测试 FastDFS

a. 在 192.168.10.11 上, 启动 tracker、nginx [root@localhost ~]# service fdfs_trackerd start [root@localhost ~]# /usr/local/nginx/sbin/nginx

b. 在 192.168.10.12-192.168.10.15 上,分别启动 storage、nginx

[root@localhost ~]# service fdfs_storaged start [root@localhost ~]# /usr/local/nginx/sbin/nginx

c. 在 192.168.10.11 上, 查看 storage 的状态

[root@localhost~]# vim /etc/fdfs/client.conf base_path=/data/fastdfs tracker_server=192.168.10.11:22122 [root@localhost~]# fdfs_monitor /etc/fdfs/client.conf

[2014-03-19 06:38:39] DEBUG - base_path=/data/fastdfs, connect_timeout=30, network_timeout=60, tracker_server_count=1, anti_steal_token=0, anti_steal_secret_key length=0, use_connection_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 192.168.10.11:22122
group count: 2
Group 1:
group name = g1
disk total space = 96558 \text{ MB}
disk free space = 84274 MB
trunk free space = 0 \text{ MB}
storage server count = 2
active server count = 2
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 = 192.168.10.12
                   ip_addr = 192.168.10.12 ACTIVE
                   http domain =
                   version = 5.01
                   join time = 2014-03-19 00:27:16
                   up time = 2014-03-19 05:47:48
                   total storage = 96558 MB
                   free storage = 85535 MB
                   upload priority = 10
                   store_path_count = 1
                   subdir_count_per_path = 256
                   storage port = 23000
                   storage http port = 8888
                   current_write_path = 0
                   source storage id=
                   if_trunk_server= 0
                   total upload count = 6
                   success_upload_count = 6
                   total append count = 0
                   success_append_count = 0
                   total_modify_count = 0
                   success modify count = 0
                   total_truncate_count = 0
                   success truncate count = 0
```

	total_set_meta_count = 6
	success_set_meta_count = 6
	total_delete_count = 0
	<pre>success_delete_count = 0</pre>
	total_download_count = 0
	<pre>success_download_count = 0</pre>
	total_get_meta_count = 0
	success_get_meta_count = 0
	total_create_link_count = 0
	success_create_link_count = 0
	total_delete_link_count = 0
	success delete link $count = 0$
	total upload bytes = 795240
	success upload bytes = 795240
	total append bytes = 0
	success append by $tes = 0$
	total modify bytes = 0
	success modify bytes = 0
	stotal download bytes = 0
	success download bytes $= 0$
	total sync in bytes = 530356
	success sync in bytes = 530356
	total_sync_out_bytes = 795534
	success_sync_out_bytes = 795534
	total file open $count = 14$
	success_file_open_count = 14
	total_file_read_count = 0
	success_file_read_count = 0
	total_file_write_count = 14
	success file write $count = 14$
	last_heart_beat_time = 2014-03-19 06:38:12
	last_source_update = 2014-03-19 00:54:09
	last sync update = $2014-03-19\ 00:54:11$
	last synced timestamp = $2014-03-19\ 00:54:11\ (0s\ delay)$
Storage 2	
	id = 192.168.10.13
	ip_addr = 192.168.10.13 ACTIVE
	http domain =
	version $= 5.01$
	join time = 2014-03-19 00:34:39
	up time = 2014-03-19 05:47:49
	total storage = 96558 MB
	free storage = 84274 MB
	upload priority $= 10$

store path count = 1subdir count per path = 256storage_port = 23000 storage_http_port = 8888 current_write_path = 0source storage id= 192.168.10.12 if trunk server=0 total upload count = 4success_upload_count = 4 total_append_count = 0success append count = 0total modify count = 0success modify count = 0total truncate count = 0success_truncate_count = 0 total set meta count = 4success set meta count = 4total delete count = 0success delete count = 0total_download_count = 0success download count = 0total_get_meta_count = 0 success get meta count = 0total_create_link_count = 0 success create link count = 0total_delete_link_count = 0 success_delete_link_count = 0 total upload bytes = 530160success upload bytes = 530160total append by tes = 0success append by tes = 0total modify bytes = 0success modify bytes = 0 $stotal_download_bytes = 0$ success download bytes = 0total_sync_in_bytes = 795534 success sync in bytes = 795534total sync out bytes = 530356success sync out bytes = 530356total_file_open_count = 16 success file open count = 16 total file read count = 0success_file_read_count = 0 total file write count = 16

```
success file write count = 16
                   last heart beat time = 2014-03-19 06:38:08
                   last_source_update = 2014-03-19 00:54:11
                   last sync update = 2014-03-19 00:54:15
                   last_synced_timestamp = 2014-03-19 00:54:09 (0s delay)
group name = g2
disk total space = 193116 MB
disk free space = 170396 MB
trunk free space = 0 \text{ MB}
storage server count = 2
active server count = 2
storage server port = 23000
storage HTTP port = 8888
store path count = 2
subdir count per path = 256
current write server index = 0
current trunk file id = 0
         Storage 1:
```

Group 2:

```
id = 192.168.10.14
ip addr = 192.168.10.14 ACTIVE
http domain =
version = 5.01
join time = 2014-03-19 01:43:00
up time = 2014-03-19 06:38:28
total storage = 193116 MB
free storage = 170480 MB
upload priority = 10
store_path_count = 2
subdir count per path = 256
storage port = 23000
storage_http_port = 8888
current write path = 0
source storage id= 192.168.10.15
if trunk server=0
total upload count = 10
success upload count = 10
total_append_count = 0
success_append_count = 0
total modify count = 0
success_modify_count = 0
total truncate count = 0
```

success truncate count = 0total set meta count = 10 $success_set_meta_count = 10$ total delete count = 0success_delete_count = 0total download count = 0success download count = 0total get meta count = 0success get meta_count = 0 total create link count = 0success create link count = 0total delete link count = 0success delete link count = 0total upload bytes = 1325400success_upload_bytes = 1325400 total append bytes = 0success append by tes = 0total modify bytes = 0success modify bytes = 0stotal download bytes = 0success download bytes = 0total_sync_in_bytes = 1325890 success sync in bytes = 1325890total sync_out_bytes = 1325890 success sync out bytes = 1325890total file open count = 30success_file_open_count = 30 total file read count = 0success file read count = 0total file write count = 30success_file_write_count = 30 last heart beat time = 2014-03-19 06:38:27 last source update = 2014-03-19 01:54:26 last sync update = 2014-03-19 01:54:28 last_synced_timestamp = 2014-03-19 01:54:27 (0s delay) Storage 2: id = 192.168.10.15 ip addr = 192.168.10.15 **ACTIVE** http domain = version = 5.01join time = 2014-03-19 01:42:41 up time = 2014-03-19 06:38:32 total storage = 193116 MB free storage = 170396 MB

upload priority = 10store path count = 2subdir_count_per_path = 256 storage port = 23000storage_http_port = 8888 current write path = 0source storage id= if trunk server=0 total_upload_count = 10 success_upload_count = 10 total append count = 0success append count = 0total modify count = 0success modify count = 0total_truncate_count = 0 success truncate count = 0total set meta count = 10success set meta count = 10total_delete_count = 0success_delete_count = 0 total download count = 0success download count = 0total get meta count = 0success get meta_count = 0 total create link count = 0success_create_link_count = 0 total_delete_link_count = 0 success delete link count = 0total upload bytes = 1325400success upload bytes = 1325400total_append_bytes = 0success append by tes = 0total modify bytes = 0 $success_modify_bytes = 0$ stotal download bytes = 0success_download_bytes = 0total sync in bytes = 1325890success sync in bytes = 1325890total_sync_out_bytes = 1325890 success_sync_out_bytes = 1325890 total_file_open_count = 30 success file open count = 30total_file_read_count = 0 success file read count = 0

total_file_write_count = 30 success_file_write_count = 30 last_heart_beat_time = 2014-03-19 06:38:32 last_source_update = 2014-03-19 01:54:27 last_sync_update = 2014-03-19 01:54:32 last_synced_timestamp = 2014-03-19 01:54:26 (0s delay)

#storage server 有 7 种状态,如下:
FDFS_STORAGE_STATUS_INIT :初始化,尚未得到同步已有数据的源服务器
FDFS_STORAGE_STATUS_WAIT_SYNC :等待同步,已得到同步已有数据的源服务器
FDFS_STORAGE_STATUS_SYNCING :同步中
FDFS_STORAGE_STATUS_DELETED :已删除
FDFS_STORAGE_STATUS_OFFLINE :离线
FDFS_STORAGE_STATUS_ONLINE :在线,尚不能提供服务
FDFS_STORAGE_STATUS_ACTIVE :在线,可以提供服务

d. 在 192.168.10.11 上, 测试文件上传

[root@localhost ~]# fdfs_upload_file /etc/fdfs/client.conf aa.jpg g1/M00/AC/2F/wKgKDVMppoGAMCFNAAIFvJcyojY165.jpg

##由于 g1 是在 192.168.10.12、192.168.10.13 机器上,所以可以通过下面地址访问 http://192.168.10.12/g1/M00/AC/2F/wKgKDVMppoGAMCFNAAIFvJcyojY165.jpg http://192.168.10.13/g1/M00/AC/2F/wKgKDVMppoGAMCFNAAIFvJcyojY165.jpg ##还可以通过 192.168.10.11 访问,建议通过 192.168.10.11 访问 http://192.168.10.11/g1/M00/AC/2F/wKgKDVMppoGAMCFNAAIFvJcyojY165.jpg