FastDFS分布式文件系统的安装与使用（单节点）

跟踪服务器：192.168.0.100 (dfs-tracker-01)

存储服务器：192.168.0.101 (dfs-storage-01)

环境：CentOS 6.6

用户：root

数据目录：/data/fastdfs（注：数据目录按你的数据盘挂载路径而定）

安装包目录： /data/software/fastdfs-packages

安装包fastdfs最新版本6.0.3：

<https://github.com/xiongshiyan/fastdfs>

<https://github.com/xiongshiyan/libfastcommon>

（是从 FastDFS和 FastDHT中提取出来的公共 C函数库）

<https://github.com/xiongshiyan/fastdfs-nginx-module>

<https://www.nginx.com/resources/wiki/>

<https://github.com/xiongshiyan/fastdfs-client-java>

源码地址：https://github.com/happyfish100/

下载地址：http://sourceforge.net/projects/fastdfs/files/

官方论坛：http://bbs.chinaunix.net/forum-240-1.html

1. 所有跟踪服务器和存储服务器均执行如下操作
2. 编译和安装所需的依赖包:

# yum install make cmake gcc gcc-c++

1. 安装 libfastcommon:
2. 上传或者下载libfastcommon-master.zip到/data/software/fastdfs-packages目录
3. 解压

# cd /data/software/fastdfs-packages

# unzip libfastcommon-master.zip

# cd libfastcommon-master

1. 编译、安装

# ./make.sh

# ./make.sh install

libfastcommon默认安装到

/usr/lib64/libfastcommon.so

/usr/lib64/libfdfsclient.so

1. 因为 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

1. 安装 FastDFS**（tracker、storage都需要安装，只是配置文件和启动脚本不一样）**
2. 上传或下载FastDFS源码包fastdfs-master.zip [6.0.3]到/data/software/fastdfs-packages目录
3. 解压

# cd /data/software/fastdfs-packages

# unzip fastdfs-master.zip

# cd fastdfs-master

1. 编译、安装（确保安装了libfastcommon）

# ./make.sh

# ./make.sh install

采用默认安装的方式安装,安装后的相应文件与目录：

服务脚本在:

/etc/init.d/fdfs\_storaged

/etc/init.d/fdfs\_trackerd

配置文件在（样例配置文件）:

/etc/fdfs/client.conf.sample

/etc/fdfs/storage.conf.sample

/etc/fdfs/tracker.conf.sample

命令工具在/usr/bin/目录下的:

/usr/bin/fdfs\_appender\_test

/usr/bin/fdfs\_appender\_test1

/usr/bin/fdfs\_append\_file

/usr/bin/fdfs\_crc32

/usr/bin/fdfs\_delete\_file

/usr/bin/fdfs\_download\_file

/usr/bin/fdfs\_file\_info

/usr/bin/fdfs\_monitor

/usr/bin/fdfs\_regenerate\_filename

/usr/bin/fdfs\_storaged

/usr/bin/fdfs\_test

/usr/bin/fdfs\_test1

/usr/bin/fdfs\_trackerd

/usr/bin/fdfs\_upload\_appender

/usr/bin/fdfs\_upload\_file

/usr/bin/restart.sh

/usr/bin/stop.sh

1. **5.0.5版本存在，6.0.3版本已修复此问题，确认即可。**~~因为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，因此需要修改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改为/usr/bin。~~

~~# vim /etc/init.d/fdfs\_storaged~~

~~使用查找替换进行统一修改： %s+/usr/local/bin+/usr/bin~~

~~# vim /etc/init.d/fdfs\_tracker~~

~~使用查找替换进行统一修改： %s+/usr/local/bin+/usr/bin~~

1. 配置fastdfs跟踪器（192.168.0.100）
2. 复制fastdfs跟踪器样例配置文件，并重命名为tracker.conf。

# cp /etc/fdfs/tracker.conf.sample /etc/fdfs/tracker.conf

1. 编辑跟踪器配置文件

# vim /etc/fdfs/tracker.conf

修改的内容如下，其他保留：

disabled=false

port=22122

base\_path=/data/fastdfs/tracker

1. 创建数据目录

# mkdir -p /data/fastdfs/tracker

1. 防火墙打开22122端口

# vim /etc/sysconfig/iptables

添加

-A INPUT -m state --state NEW -m tcp -p tcp --dport 22122 -j ACCEPT

重启

# service iptables restart

1. 启动tracker

# /etc/init.d/fdfs\_trackerd start

初次启动成功，会在/data/fastdfs/tracker 目录创建data、logs两个目录

1. 关闭tracker

# /etc/init.d/fdfs\_trackerd stop

1. 设置tracker开机启动

# vim /etc/rc.d/rc.local

/etc/init.d/fdfs\_trackerd start

1. 配置fastdfs存储器（192.168.0.101）
2. 复制fastdfs存储器样例配置文件重命名为storage.conf

# cp /etc/fdfs/storage.conf.sample /etc/fdfs/storage.conf

1. 编辑存储器配置文件

# vim /etc/fdfs/storage.conf

修改的内容如下，其他保留

disable=false

port=23000

base\_path=/data/fastdfs/storage

store\_path0=/data/fastdfs/storage

tracker\_server=192.168.0.100:22122

http.server\_port=8888

1. 创建数据目录

# mkdir -p /data/fastdfs/storage

1. 防火墙打开23000端口

# vim /etc/sysconfig/iptables

添加

-A INPUT -m state --state NEW -m tcp -p tcp --dport 23000 -j ACCEPT

重启

# service iptables restart

1. 启动storage

# /etc/init.d/fdfs\_storaged start

初次启动成功，会在/data/fastdfs/storage 目录创建data、logs两个目录

1. 关闭storage

# /etc/init.d/fdfs\_storaged stop

1. 设置storage开机启动

# vim /etc/rc.d/rc.local

/etc/init.d/fdfs\_storaged start

1. 文件上传测试（192.168.0.100）
2. 修改tracker服务器中的客户端配置文件

# cp /etc/fdfs/client.conf.sample /etc/fdfs/client.conf

# vim /etc/fdfs/client.conf

修改

base\_path=/data/fastdfs/tracker

tracker\_server=192.168.0.100.22122

1. 执行文件上传命令

# /usr/bin/fdfs\_upload\_file /etc/fdfs/client.conf XXXXXXXXXX

返回ID号则成功： group1/M00/00/00/Ci9fJF3XZ6KADez4AA28tsAei6c591.zip

1. 在storage节点上安装nginx（192.168.0.101）
2. fastdfs-nginx-module作用

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入

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 192.168.0.101，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 192.168.0.102，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 192.168.0.102 上取文件,就会出现文件无法访问的 错误。而 fastdfs-nginx-module 可以重定向文件连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。（解压后的 fastdfs-nginx-module 在 nginx 安装时使用）

1. 上传fastdfs-nginx-module-master.zip到/data/software/fastdfs-packages
2. 解压

# cd /data/software/fastdfs-packages

# unzip fastdfs-nginx-module-master.zip

~~# cd fastdfs-nginx-module-master/src~~

~~# vim vi config~~

~~CORE\_INCS="$CORE\_INCS /usr/local/include/fastdfs /usr/local/include/fastcommon/"~~

~~修改为：~~

~~CORE\_INCS="$CORE\_INCS /usr/include/fastdfs /usr/include/fastcommon/"~~

~~（注意：这个路径修改是很重要的，不然在 nginx 编译的时候会报错的）~~

**5.0.5版本存在，6.0.3版本已修复此问题，确认即可**

1. 上传nginx稳定版本nginx-1.12.2.tar.gz到/data/software/fastdfs-packages
2. 安装编译 Nginx 所需的依赖包

# yum install gcc gcc-c++ make automake autoconf libtool pcre\* zlib openssl openssl-devel

1. 编译安装nginx，安装到/data/nginx-fastdfs,添加fastdfs-nginx-module模块

# cd /data/software/fastdfs-packages

# tar -zxvf nginx-1.12.2.tar.gz

# cd nginx-1.12.2

# ./configure --add-module=/data/software/fastdfs-packages/fastdfs-nginx-module-master/src --prefix=/data/nginx-fastdfs

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

# cp /data/software/fastdfs-packages/fastdfs-nginx-module-master/src/mod\_fastdfs.conf /etc/fdfs/

# vim /etc/fdfs/mod\_fastdfs.conf

修改如下配置，其他保留

connect\_timeout=10

base\_path=/tmp

tracker\_server=192.168.0.100:22122

storage\_server\_port=23000

group\_name=group1

url\_have\_group\_name = true

store\_path0=/data/fastdfs/storage

1. 复制fastdfs的部分配置文件到/etc/fdfs目录

# cd /data/software/fastdfs-packages/fastdfs-master/conf

# cp http.conf mime.types /etc/fdfs/

1. 在/data/fastdfs/storage目录创建软连接，将其链接到数据的实际目录

# ln -s /data/fastdfs/storage/data/ /fastdfs/storage/data/M00

1. Nginx配置

user root;

worker\_processes 1;

events {

worker\_connections 1024;

}

http {

include mime.types;

default\_type application/octet-stream;

sendfile on;

keepalive\_timeout 65;

server {

listen 8888;

server\_name localhost;

location / {

root html;

index index.html index.htm;

}

location ~/group([0-9])/M00 {

ngx\_fastdfs\_module;

}

error\_page 500 502 503 504 /50x.html;

location = /50x.html {

root html;

}

}

}

1. 防火墙打开8888端口

# vi /etc/sysconfig/iptables

添加：

-A INPUT -m state --state NEW -m tcp -p tcp --dport 8888 -j ACCEPT

# service iptables restart

1. 启动nginx

# /data/nginx-fastdfs/sbin/nginx

ngx\_http\_fastdfs\_set pid=xxx

1. 设置nginx开机启动

# vim /etc/rc.d/rc.local

/data/nginx-fastdfs/sbin/nginx

通过浏览器访问刚刚测试上传的文件

http://192.168.0.101:8888/group1/M00/00/00/Ci9fJF3XZ6KADez4AA28tsAei6c591.zip