

<http://bbs.chinaunix.net/thread-1941456-1-1.html>

首先是 **tracker.conf**

```
# is this config file disabled
# false for enabled
# true for disabled
disabled=false
# 这个配置文件是否不生效,呵呵(改成是否生效是不是会让人感觉好点呢?) false 为生效(否则不生效) true 反之

# bind an address of this host
# empty for bind all addresses of this host
bind_addr=
# 是否绑定 IP,
# bind_addr= 后面为绑定的 IP 地址 (常用于服务器有多个 IP 但只希望一个 IP 提供服务)。如果不填则表示所有的(一般不填就 OK),相信较熟练的 SA 都常用到类似功能,很多系统和应用都有

# the tracker server port
port=22122
# 提供服务的端口,不作过多解释了

# connect timeout in seconds
# default value is 30s
connect_timeout=30
#连接超时时间,针对 socket 套接字函数 connect

# network timeout in seconds
network_timeout=60
# tracker server 的网络超时,单位为秒。发送或接收数据时,如果在超时时间后还不能发送或接收数据,则本次网络通信失败。

# the base path to store data and log files
base_path=/home/yuqing/fastdfs
# base_path 目录地址(根目录必须存在,子目录会自动创建)
# 附目录说明:
# tracker server 目录及文件结构:
${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

数据文件 `storage_groups.dat` 和 `storage_servers.dat` 中的记录之间以换行符 (`\n`) 分隔, 字段之间以西文逗号 (`,`) 分隔。

`storage_groups.dat` 中的字段依次为:

1. `group_name`: 组名
2. `storage_port`: storage server 端口号

`storage_servers.dat` 中记录 storage server 相关信息, 字段依次为:

1. `group_name`: 所属组名
2. `ip_addr`: ip 地址
3. `status`: 状态
4. `sync_src_ip_addr`: 向该 storage server 同步已有数据文件的源服务器
5. `sync_until_timestamp`: 同步已有数据文件的截至时间 (UNIX 时间戳)
6. `stat.total_upload_count`: 上传文件次数
7. `stat.success_upload_count`: 成功上传文件次数
8. `stat.total_set_meta_count`: 更改 meta data 次数
9. `stat.success_set_meta_count`: 成功更改 meta data 次数
10. `stat.total_delete_count`: 删除文件次数
11. `stat.success_delete_count`: 成功删除文件次数
12. `stat.total_download_count`: 下载文件次数
13. `stat.success_download_count`: 成功下载文件次数
14. `stat.total_get_meta_count`: 获取 meta data 次数
15. `stat.success_get_meta_count`: 成功获取 meta data 次数
16. `stat.last_source_update`: 最近一次源头更新时间 (更新操作来自客户端)
17. `stat.last_sync_update`: 最近一次同步更新时间 (更新操作来自其他 storage server 的同步)

max concurrent connections this server supported

max_connections worker threads start when this service startup

`max_connections=256`

系统提供服务时的最大连接数。对于 V1.x, 因一个连接由一个线程服务, 也就是工作线程数。

对于 V2.x, 最大连接数和工作线程数没有任何关系

work thread count, should <= max_connections

default value is 4

since V2.00

V2.0 引入的这个参数, 工作线程数, 通常设置为 CPU 数

`work_threads=4`

the method of selecting group to upload files

0: round robin

1: specify group



龙果学院微信公众号: ron-coo

```
# 2: load balance, select the max free space group to upload file
store_lookup=2
# 上传组(卷) 的方式 0:轮询方式 1: 指定组 2: 平衡负载(选择最大剩余空间的组(卷)上传)
# 这里如果在应用层指定了上传到一个固定组,那么这个参数被绕过

# which group to upload file
# when store_lookup set to 1, must set store_group to the group name
store_group=group2
# 当上一个参数设定为 1 时 (store_lookup=1, 即指定组名时), 必须设置本参数为系统中存在的一个组名。如果选择其他的上传方式, 这个参数就没有效了。

# which storage server to upload file
# 0: round robin (default)
# 1: the first server order by ip address
# 2: the first server order by priority (the minimal)
store_server=0
# 选择哪个 storage server 进行上传操作(一个文件被上传后, 这个 storage server 就相当于这个文件的 storage server 源, 会对同组的 storage server 推送这个文件达到同步效果)
# 0: 轮询方式
# 1: 根据 ip 地址进行排序选择第一个服务器 (IP 地址最小者)
# 2: 根据优先级进行排序 (上传优先级由 storage server 来设置, 参数名为 upload_priority)

# which path(means disk or mount point) of the storage server to upload file
# 0: round robin
# 2: load balance, select the max free space path to upload file
store_path=0
# 选择 storage server 中的哪个目录进行上传。storage server 可以有多个存放文件的 base path (可以理解为多个磁盘)。
# 0: 轮流方式, 多个目录依次存放文件
# 2: 选择剩余空间最大的目录存放文件 (注意: 剩余磁盘空间是动态的, 因此存储到的目录或磁盘可能也是变化的)

# which storage server to download file
# 0: round robin (default)
# 1: the source storage server which the current file uploaded to
download_server=0
# 选择哪个 storage server 作为下载服务器
# 0: 轮询方式, 可以下载当前文件的任一 storage server
# 1: 哪个为源 storage server 就用哪一个 (前面说过了这个 storage server 源 是怎样产生的) 就是之前上传到哪个 storage server 服务器就是哪个了

# reserved storage space for system or other applications.
# if the free(available) space of any stoarge server in
# a group <= reserved_storage_space,
```



```
# no file can be uploaded to this group.
# bytes unit can be one of follows:
### G or g for gigabyte(GB)
### M or m for megabyte(MB)
### K or k for kilobyte(KB)
### no unit for byte(B)
### XX.XX% as ratio such as reserved_storage_space = 10%
reserved_storage_space = 10%
# storage server 上保留的空间, 保证系统或其他应用需求空间。可以用绝对值或者百分比
  (V4 开始支持百分比方式)。
#(指出 如果同组的服务器的硬盘大小一样,以最小的为准,也就是只要同组中有一台服务器
  达到这个标准了,这个标准就生效,原因就是因为他们进行备份)

#standard log level as syslog, case insensitive, value list:
### emerg for emergency
### alert
### crit for critical
### error
### warn for warning
### notice
### info
### debug
log_level=info
# 选择日志级别(日志写在哪儿?看前面的说明了,有目录介绍哦 呵呵)

#unix group name to run this program,
#not set (empty) means run by the group of current user
run_by_group=
# 操作系统运行 FastDFS 的用户组 (不填 就是当前用户组,哪个启动进程就是哪个)

#unix username to run this program,
#not set (empty) means run by current user
run_by_user=
# 操作系统运行 FastDFS 的用户 (不填 就是当前用户,哪个启动进程就是哪个)

# allow_hosts can occur more than once, host can be hostname or ip address,
# "*" means match all ip addresses, can use range like this: 10.0.1.[1-15,20] or
# host[01-08,20-25].domain.com, for example:
# allow_hosts=10.0.1.[1-15,20]
# allow_hosts=host[01-08,20-25].domain.com
allow_hosts=*
# 可以连接到此 tracker server 的 ip 范围(对所有类型的连接都有影响,包括客户端, storage
  server)
```



```
# sync log buff to disk every interval seconds
# default value is 10 seconds
sync_log_buff_interval = 10
# 同步或刷新日志信息到硬盘的时间间隔，单位为秒
# 注意：tracker server 的日志不是时时写硬盘的，而是先写内存。

# check storage server alive interval
check_active_interval = 120
# 检测 storage server 存活的时间间隔，单位为秒。
# storage server 定期向 tracker server 发心跳，如果 tracker server 在一个 check_active_interval
# 内还没有收到 storage server 的一次心跳，那边将认为该 storage server 已经下线。所以本参
# 数值必须大于 storage server 配置的心跳时间间隔。通常配置为 storage server 心跳时间间隔
# 的 2 倍或 3 倍。

# thread stack size, should > 512KB
# default value is 1MB
thread_stack_size=1MB
# 线程栈的大小。FastDFS server 端采用了线程方式。更正一下，tracker server 线程栈不应小
# 于 64KB，不是 512KB。
# 线程栈越大，一个线程占用的系统资源就越多。如果要启动更多的线程（V1.x 对应的参数
# 为 max_connections，
# V2.0 为 work_threads），可以适当降低本参数值。

# auto adjust when the ip address of the storage server changed
# default value is true
storage_ip_changed_auto_adjust=true
# 这个参数控制当 storage server IP 地址改变时，集群是否自动调整。注：只有在 storage
# server 进程重启时才完成自动调整。

# storage sync file max delay seconds
# default value is 86400 seconds (one day)
# since V2.0.0
storage_sync_file_max_delay = 86400
# V2.0 引入的参数。存储服务器之间同步文件的最大延迟时间，缺省为 1 天。根据实际情况
# 进行调整
# 注：本参数并不影响文件同步过程。本参数仅在下载文件时，判断文件是否已经被同步完
# 成的一个阈值（经验值）

# the max time of storage sync a file
# default value is 300 seconds
# since V2.0.0
storage_sync_file_max_time = 300
# V2.0 引入的参数。存储服务器同步一个文件需要消耗的最大时间，缺省为 300s，即 5 分
# 钟。
```



注: 本参数并不影响文件同步过程。本参数仅在下载文件时, 作为判断当前文件是否被同步完成的一个阈值 (经验值)

if use a trunk file to store several small files

default value is false

since V3.00

use_trunk_file = false

V3.0 引入的参数。是否使用小文件合并存储特性, 缺省是关闭的。

the min slot size, should <= 4KB

default value is 256 bytes

since V3.00

slot_min_size = 256

V3.0 引入的参数。

trunk file 分配的最小字节数。比如文件只有 16 个字节, 系统也会分配 slot_min_size 个字节。

the max slot size, should > slot_min_size

store the upload file to trunk file when it's size <= this value

default value is 16MB

since V3.00

slot_max_size = 16MB

V3.0 引入的参数。

只有文件大小<=这个参数值的文件, 才会合并存储。如果一个文件的大小大于这个参数值, 将直接保存到一个文件中 (即不采用合并存储方式)。

the trunk file size, should >= 4MB

default value is 64MB

since V3.00

trunk_file_size = 64MB

V3.0 引入的参数。

合并存储的 trunk file 大小, 至少 4MB, 缺省值是 64MB。不建议设置得过大。

if create trunk file advancely

default value is false

trunk_create_file_advance = false

是否提前创建 trunk file。只有当这个参数为 true, 下面 3 个以 trunk_create_file_打头的参数才有效。

the time base to create trunk file

the time format: HH:MM

default value is 02:00

trunk_create_file_time_base = 02:00

提前创建 trunk file 的起始时间点 (基准时间), 02:00 表示第一次创建的时间点是凌晨 2



点。

```
# the interval of create trunk file, unit: second
# default value is 38400 (one day)
trunk_create_file_interval = 86400
# 创建 trunk file 的时间间隔, 单位为秒。如果每天只提前创建一次, 则设置为 86400

# the threshold to create trunk file
# when the free trunk file size less than the threshold, will create
# the trunk files
# default value is 0
trunk_create_file_space_threshold = 20G
# 提前创建 trunk file 时, 需要达到的空闲 trunk 大小
# 比如本参数为 20G, 而当前空闲 trunk 为 4GB, 那么只需要创建 16GB 的 trunk file 即可。

# if check trunk space occupying when loading trunk free spaces
# the occupied spaces will be ignored
# default value is false
# since V3.09
# NOTICE: set this parameter to true will slow the loading of trunk spaces
# when startup. you should set this parameter to true when neccessary.
trunk_init_check_occupying = false
#trunk 初始化时, 是否检查可用空间是否被占用

# if ignore storage_trunk.dat, reload from trunk binlog
# default value is false
# since V3.10
# set to true once for version upgrade when your version less than V3.10
trunk_init_reload_from_binlog = false
# 是否无条件从 trunk binlog 中加载 trunk 可用空间信息
# FastDFS 缺省是从快照文件 storage_trunk.dat 中加载 trunk 可用空间,
# 该文件的第一行记录的是 trunk binlog 的 offset, 然后从 binlog 的 offset 开始加载

# if use storage ID instead of IP address
# default value is false
# since V4.00
use_storage_id = false
# 是否使用 server ID 作为 storage server 标识

# specify storage ids filename, can use relative or absolute path
# since V4.00
storage_ids_filename = storage_ids.conf
# use_storage_id 设置为 true, 才需要设置本参数
# 在文件中设置组名、server ID 和对应的 IP 地址, 参见源码目录下的配置示例:
```



conf/storage_ids.conf

```
# if store slave file use symbol link
# default value is false
# since V4.01
store_slave_file_use_link = false
# 存储从文件是否采用 symbol link（符号链接）方式
# 如果设置为 true，一个从文件将占用两个文件：原始文件及指向它的符号链接。

# if rotate the error log every day
# default value is false
# since V4.02
rotate_error_log = false
# 是否定期轮转 error log，目前仅支持一天轮转一次

# rotate error log time base, time format: Hour:Minute
# Hour from 0 to 23, Minute from 0 to 59
# default value is 00:00
# since V4.02
error_log_rotate_time=00:00
# error log 定期轮转的时间点，只有当 rotate_error_log 设置为 true 时有效

# rotate error log when the log file exceeds this size
# 0 means never rotates log file by log file size
# default value is 0
# since V4.02
rotate_error_log_size = 0
# error log 按大小轮转
# 设置为 0 表示不按文件大小轮转，否则当 error log 达到该大小，就会轮转到新文件中

# 以下是关于 http 的设置了 默认编译是不生效的 要求更改 #WITH_HTTPD=1 将 注释#去掉 再编译
# 关于 http 的应用 说实话 不是很了解 没有见到 相关说明 ,望 版主可以完善一下 以下是字面解释了
#HTTP settings
http.disabled=false    # HTTP 服务是否不生效
http.server_port=8080  # HTTP 服务端口

#use "#include" directive to include http other settiongs
##include http.conf    # 如果加载 http.conf 的配置文件 去掉第一个#
```

哈哈 完成了一个 下面是 storage.conf




```
# is this config file disabled
# false for enabled
# true for disabled
disabled=false
#同上文了 就不多说了

# the name of the group this storage server belongs to
group_name=group1
# 指定 此 storage server 所在 组(卷)

# bind an address of this host
# empty for bind all addresses of this host
bind_addr=
# 同上文

# if bind an address of this host when connect to other servers
# (this storage server as a client)
# true for binding the address configed by above parameter: "bind_addr"
# false for binding any address of this host
client_bind=true
# bind_addr 通常是针对 server 的。当指定 bind_addr 时，本参数才有效。
# 本 storage server 作为 client 连接其他服务器（如 tracker server、其他 storage server），是否绑定 bind_addr。

# the storage server port
port=23000
# storage server 服务端口

# connect timeout in seconds
# default value is 30s
connect_timeout=30
#连接超时时间，针对 socket 套接字函数 connect

# network timeout in seconds
network_timeout=60
# storage server 网络超时时间，单位为秒。发送或接收数据时，如果在超时时间后还不能发送或接收数据，则本次网络通信失败。

# heart beat interval in seconds
heart_beat_interval=30
# 心跳间隔时间，单位为秒 (这里是指主动向 tracker server 发送心跳)

# disk usage report interval in seconds
```



stat_report_interval=60

storage server 向 tracker server 报告磁盘剩余空间的时间间隔, 单位为秒。

the base path to store data and log files

base_path=/home/yuqing/fastdfs

base_path 目录地址,根目录必须存在 子目录会自动生成 (注 :这里不是上传的文件存放的地址,之前是的,在某个版本后更改了)

目录结构 因为 版主没有更新到 论坛上 这里就不发了 大家可以看一下置顶贴:

max concurrent connections server supported

max_connections worker threads start when this service startup

max_connections=256

同上文

work thread count, should <= max_connections

default value is 4

since V2.00

V2.0 引入的这个参数, 工作线程数, 通常设置为 CPU 数

work_threads=4

the buff size to recv / send data

default value is 64KB

since V2.00

buff_size = 256KB

V2.0 引入本参数。设置队列结点的 buffer 大小。工作队列消耗的内存大小 = buff_size * max_connections

设置得大一些, 系统整体性能会有所提升。

消耗的内存请不要超过系统物理内存大小。另外, 对于 32 位系统, 请注意使用到的内存不要超过 3GB

if read / write file directly

if set to true, open file will add the O_DIRECT flag to avoid file caching

by the file system. be careful to set this parameter.

default value is false

disk_rw_direct = false

V2.09 引入本参数。设置为 true, 表示不使用操作系统的文件内容缓冲特性。

如果文件数量很多, 且访问很分散, 可以考虑将本参数设置为 true

if disk read / write separated

false for mixed read and write

true for separated read and write

default value is true

since V2.00



disk_rw_separated = true

V2.0 引入本参数。磁盘 IO 读写是否分离，缺省是分离的。

disk reader thread count per store base path

for mixed read / write, this parameter can be 0

default value is 1

since V2.00

disk_reader_threads = 1

V2.0 引入本参数。针对单个存储路径的读线程数，缺省值为 1。

读写分离时，系统中的读线程数 = disk_reader_threads * store_path_count

读写混合时，系统中的读写线程数 = (disk_reader_threads + disk_writer_threads) * store_path_count

disk writer thread count per store base path

for mixed read / write, this parameter can be 0

default value is 1

since V2.00

disk_writer_threads = 1

V2.0 引入本参数。针对单个存储路径的写线程数，缺省值为 1。

读写分离时，系统中的写线程数 = disk_writer_threads * store_path_count

读写混合时，系统中的读写线程数 = (disk_reader_threads + disk_writer_threads) * store_path_count

when no entry to sync, try read binlog again after X milliseconds

0 for try again immediately (not need to wait)

sync_wait_msec=200

同步文件时，如果从 binlog 中没有读到要同步的文件，休眠 N 毫秒后重新读取。0 表示不休眠，立即再次尝试读取。

出于 CPU 消耗考虑，不建议设置为 0。如何希望同步尽可能快一些，可以将本参数设置得小一些，比如设置为 10ms

after sync a file, usleep milliseconds

0 for sync successively (never call usleep)

sync_interval=0

同步上一个文件后，再同步下一个文件的时间间隔，单位为毫秒，0 表示不休眠，直接同步下一个文件。

sync start time of a day, time format: Hour:Minute

Hour from 0 to 23, Minute from 0 to 59

sync_start_time=00:00

sync end time of a day, time format: Hour:Minute

Hour from 0 to 23, Minute from 0 to 59

sync_end_time=23:59



上面二个一起解释。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, 相信 sa 都会明白

write to the mark file after sync N files

default value is 500

write_mark_file_freq=500

同步完 N 个文件后, 把 storage 的 mark 文件同步到磁盘

注: 如果 mark 文件内容没有变化, 则不会同步

path(disk or mount point) count, default value is 1

store_path_count=1

存放文件时 storage server 支持多个路径 (例如磁盘)。这里配置存放文件的基路径数目, 通常只配一个目录。

store_path#, based 0, if store_path0 not exists, it's value is base_path

the paths must be exist

store_path0=/home/yuqing/fastdfs

#store_path1=/home/yuqing/fastdfs2

逐一配置 store_path 个路径, 索引号基于 0。注意配置方法后面有 0,1,2, 需要配置 0 到 store_path - 1。

如果不配置 base_path0, 那边它就和 base_path 对应的路径一样。

subdir_count * subdir_count directories will be auto created under each

store_path (disk), value can be 1 to 256, default value is 256

subdir_count_per_path=256

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 (系统的存储机制, 大家看看文件存储的目录就知道了)

如果本参数只为 N (如: 256), 那么 storage server 在初次运行时, 会自动创建 N * N 个存放文件的子目录。

tracker_server can occur more than once, and tracker_server format is

"host:port", host can be hostname or ip address

tracker_server=10.62.164.84:22122

tracker_server=10.62.245.170:22122

tracker_server 的列表 要写端口的哦 (再次提醒是主动连接 tracker_server)

有多个 tracker server 时, 每个 tracker server 写一行

#standard log level as syslog, case insensitive, value list:

emerg for emergency

alert

crit for critical

error

warn for warning

notice



```
### info
### debug
log_level=info
# 日志级别不多说

#unix group name to run this program,
#not set (empty) means run by the group of current user
run_by_group=
# 同上文了

#unix username to run this program,
#not set (empty) means run by current user
run_by_user=
# 同上文了 (提醒注意权限 如果和 webserver 不搭 可以会产生错误 哦)

# allow_hosts can occur more than once, host can be hostname or ip address,
# "*" means match all ip addresses, can use range like this: 10.0.1.[1-15,20] or
# host[01-08,20-25].domain.com, for example:
# allow_hosts=10.0.1.[1-15,20]
# allow_hosts=host[01-08,20-25].domain.com
allow_hosts=*
# 允许连接本 storage server 的 IP 地址列表 (不包括自带 HTTP 服务的所有连接)
# 可以配置多行, 每行都会起作用

# the mode of the files distributed to the data path
# 0: round robin(default)
# 1: random, distributed by hash code
file_distribute_path_mode=0
# 文件在 data 目录下分散存储策略。
# 0: 轮流存放, 在一个目录下存储设置的文件数后 (参数 file_distribute_rotate_count 中设置文件数), 使用下一个目录进行存储。
# 1: 随机存储, 根据文件名对应的 hash code 来分散存储。

# valid when file_distribute_to_path is set to 0 (round robin),
# when the written file count reaches this number, then rotate to next path
# default value is 100
file_distribute_rotate_count=100
# 当上面的参数 file_distribute_path_mode 配置为 0 (轮流存放方式) 时, 本参数有效。
# 当一个目录下的文件存放的文件数达到本参数值时, 后续上传的文件存储到下一个目录中。

# call fsync to disk when write big file
# 0: never call fsync
# other: call fsync when written bytes >= this bytes
```



```
# default value is 0 (never call fsync)
fsync_after_written_bytes=0
# 当写入大文件时, 每写入 N 个字节, 调用一次系统函数 fsync 将内容强行同步到硬盘。0
表示从不调用 fsync

# sync log buff to disk every interval seconds
# default value is 10 seconds
sync_log_buff_interval=10
# 同步或刷新日志信息到硬盘的时间间隔, 单位为秒
# 注意: storage server 的日志信息不是时时写硬盘的, 而是先写内存。

# sync binlog buff / cache to disk every interval seconds
# this parameter is valid when write_to_binlog set to 1
# default value is 60 seconds
sync_binlog_buff_interval=60
# 同步 binlog (更新操作日志) 到硬盘的时间间隔, 单位为秒
# 本参数会影响新上传文件同步延迟时间

# sync storage stat info to disk every interval seconds
# default value is 300 seconds
sync_stat_file_interval=300
# 把 storage 的 stat 文件同步到磁盘的时间间隔, 单位为秒。
# 注: 如果 stat 文件内容没有变化, 不会进行同步

# thread stack size, should >= 512KB
# default value is 512KB
thread_stack_size=512KB
# 线程栈的大小。FastDFS server 端采用了线程方式。
# 对于 V1.x, storage server 线程栈不应小于 512KB; 对于 V2.0, 线程栈大于等于 128KB 即可。
# 线程栈越大, 一个线程占用的系统资源就越多。
# 对于 V1.x, 如果要启动更多的线程 (max_connections), 可以适当降低本参数值。

# the priority as a source server for uploading file.
# the lower this value, the higher its uploading priority.
# default value is 10
upload_priority=10
# 本 storage server 作为源服务器, 上传文件的优先级, 可以为负数。值越小, 优先级越高。
这里就和 tracker.conf 中 store_server= 2 时的配置相对应了

# if check file duplicate, when set to true, use FastDHT to store file indexes
# 1 or yes: need check
# 0 or no: do not check
# default value is 0
```



check_file_duplicate=0

是否检测上传文件已经存在。如果已经存在, 则不存在文件内容, 建立一个符号链接以节省磁盘空间。

这个应用要配合 FastDHT 使用, 所以打开前要先安装 FastDHT

1 或 yes 是检测, 0 或 no 是不检测

file signature method for check file duplicate

hash: four 32 bits hash code

md5: MD5 signature

default value is hash

since V4.01

file_signature_method=hash

文件去重时, 文件内容的签名方式:

hash: 4 个 hash code

md5: MD5

namespace for storing file indexes (key-value pairs)

this item must be set when check_file_duplicate is true / on

key_namespace=FastDFS

当上个参数设定为 1 或 yes 时 (true/on 也是可以的), 在 FastDHT 中的命名空间。

set keep_alive to 1 to enable persistent connection with FastDHT servers

default value is 0 (short connection)

keep_alive=0

与 FastDHT servers 的连接方式 (是否为持久连接), 默认是 0 (短连接方式)。可以考虑使用长连接, 这要看 FastDHT server 的连接数是否够用。

下面是关于 FastDHT servers 的设定 需要对 FastDHT servers 有所了解, 这里只说字面意思了

you can use "#include filename" (not include double quotes) directive to

load FastDHT server list, when the filename is a relative path such as

pure filename, the base path is the base path of current/this config file.

must set FastDHT server list when check_file_duplicate is true / on

please see INSTALL of FastDHT for detail

##include /home/youqing/fastdht/conf/fdht_servers.conf

可以通过 #include filename 方式来加载 FastDHT servers 的配置, 装上 FastDHT 就知道该如何配置啦。

同样要求 check_file_duplicate=1 时才有用, 不然系统会忽略

fdht_servers.conf 记载的是 FastDHT servers 列表

if log to access log

default value is false

since V4.00

use_access_log = false



```
# 是否将文件操作记录到 access log

# if rotate the access log every day
# default value is false
# since V4.00
rotate_access_log = false
# 是否定期轮转 access log, 目前仅支持一天轮转一次

# rotate access log time base, time format: Hour:Minute
# Hour from 0 to 23, Minute from 0 to 59
# default value is 00:00
# since V4.00
access_log_rotate_time=00:00
# access log 定期轮转的时间点, 只有当 rotate_access_log 设置为 true 时有效

# if rotate the error log every day
# default value is false
# since V4.02
rotate_error_log = false
# 是否定期轮转 error log, 目前仅支持一天轮转一次

# rotate error log time base, time format: Hour:Minute
# Hour from 0 to 23, Minute from 0 to 59
# default value is 00:00
# since V4.02
error_log_rotate_time=00:00
# error log 定期轮转的时间点, 只有当 rotate_error_log 设置为 true 时有效

# rotate access log when the log file exceeds this size
# 0 means never rotates log file by log file size
# default value is 0
# since V4.02
rotate_access_log_size = 0
# access log 按文件大小轮转
# 设置为 0 表示不按文件大小轮转, 否则当 access log 达到该大小, 就会轮转到新文件中

# rotate error log when the log file exceeds this size
# 0 means never rotates log file by log file size
# default value is 0
# since V4.02
rotate_error_log_size = 0
# error log 按文件大小轮转
# 设置为 0 表示不按文件大小轮转, 否则当 error log 达到该大小, 就会轮转到新文件中
```



```
# if skip the invalid record when sync file
# default value is false
# since V4.02
file_sync_skip_invalid_record=false
# 文件同步的时候, 是否忽略无效的 binlog 记录
```

下面是 http 的配置了。如果系统较大, 这个服务有可能支持不了, 可以自行换一个 webserver, 我喜欢 lighttpd, 当然 ng 也很好了。具体不说明了。相应这一块的说明大家都懂, 不明白见上文。

```
#HTTP settings
http.disabled=false
```

```
# the port of the web server on this storage server
http.server_port=8888
```

```
http.trunk_size=256KB
# http.trunk_size 表示读取文件内容的 buffer 大小 (一次读取的文件内容大小), 也就是回复
给 HTTP client 的块大小。
```

```
# use the ip address of this storage server if domain_name is empty,
# else this domain name will occur in the url redirected by the tracker server
http.domain_name=
# storage server 上 web server 域名, 通常仅针对单独部署的 web server。这样 URL 中就可以
通过域名方式来访问 storage server 上的文件了,
# 这个参数为空就是 IP 地址的方式。
```

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



补充:

storage.conf 中影响 storage server 同步速度的参数有如下几个:

when no entry to sync, try read binlog again after X milliseconds

0 for try again immediately (not need to wait)

sync_wait_msec=200

同步文件时, 如果从 binlog 中没有读到要同步的文件, 休眠 N 毫秒后重新读取。0 表示不休眠, 立即再次尝试读取。

不建议设置为 0, 如何希望同步尽可能快一些, 可以将本参数设置得小一些, 比如设置为 10ms

after sync a file, usleep milliseconds

0 for sync successively (never call usleep)

sync_interval=0

同步上一个文件后, 再同步下一个文件的时间间隔, 单位为毫秒, 0 表示不休眠, 直接同步下一个文件。

sync start time of a day, time format: Hour:Minute

Hour from 0 to 23, Minute from 0 to 59

sync_start_time=00:00

sync end time of a day, time format: Hour:Minute

Hour from 0 to 23, Minute from 0 to 59

sync_end_time=23:59

上面二个一起解释。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, 相信 sa 都会明白

sync binlog buff / cache to disk every interval seconds

this parameter is valid when write_to_binlog set to 1

default value is 60 seconds

sync_binlog_buff_interval=60

同步 binlog (更新操作日志) 到硬盘的时间间隔, 单位为秒

本参数会影响新上传文件同步延迟时间

