flatq安装手册

**(V1.0)**

*变更记录*

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **状态** | **描述** | **作者** |
| 2015-01-17 | V0.1 | 初稿 | denglei |

目录

[1 编译、安装与配置 4](#_Toc410286707)

[1.1 依赖软件包 4](#_Toc410286708)

[1.2 依赖包安装编译 4](#_Toc410286709)

[1.2.1 Mysql client lib的安装 4](#_Toc410286710)

[1.2.2 Openssl 开发库 4](#_Toc410286711)

[1.2.3 Expat编译安装 4](#_Toc410286712)

[1.2.4 Cwinux编译安装 5](#_Toc410286713)

[1.3 mq编译安装 5](#_Toc410286714)

[1.4 配置与启动 5](#_Toc410286715)

[1.4.1 flatq的配置与启动 5](#_Toc410286716)

[1.4.2 product\_proxy(写代理)的配置与启动 8](#_Toc410286717)

[1.4.3 subscribe\_proxy (读代理)的配置与启动 9](#_Toc410286718)

[2 测试与接口 10](#_Toc410286719)

[2.1 发送消息 10](#_Toc410286720)

[2.2 同步消息 10](#_Toc410286721)

[2.3 接口 11](#_Toc410286722)

## 编译、安装与配置

### 依赖软件包

依赖如下的软件包：

* mysql client lib：版本不低于5.0
* openssl开发库: 版本不低于0.8
* expat： 版本不低于2.0
* cwinux : 版本不低于2.4.1

### 依赖包安装编译

#### Mysql client lib的安装

#下面是centos64环境下的rpm安装，根据自己实际情况决定是否安装

#yum install mysql-devel.x86\_64

#### Openssl 开发库

#下面是centos64环境下的rpm安装，根据自己实际情况决定是否安装

#yum install openssl-devel.x86\_64

#### Expat编译安装

* 下载软件包：

#wget <http://jaist.dl.sourceforge.net/project/expat/expat/2.1.0/expat-2.1.0.tar.gz>

* 解压编译安装：

#tar xvfz expat-2.1.0.tar.gz

#cd expat-2.1.0

#./configure --enable-shared=no ##形成静态库

#make

#make install

#### Cwinux编译安装

* 下载软件包：

#wget https://github.com/dengleitju/cwinux/archive/v2.4.1.tar.gz -O cwinux.2.4.1.tar.gz

* 解压编译安装：

#tar xvfz cwinux.2.4.1.tar.gz

#cd cwinux

##########形成静态库

#./configure --prefix=/usr/local/cwinux --enable-shared=no

#make

#make install

### mq编译安装

* 下载软件包：

# wget<https://github.com/dengleitju/flatq/archive/v1.0.2.tar.gz> -O flatq.1.0.2.tar.gz

* 解压安装：

#tar xvfz flatq.1.0.2.tar.gz

#cd flatq-1.0.2

#export LD\_LIBRARY\_PATH=$LD\_LIBRARY\_PATH:/usr/local/cwinux/lib/

* 编译安装flatq：

#./configure --with-cwinux=/usr/local/cwinux/ --with-zk=/usr/local/ --with-openssl=/usr/

#make

### 配置与启动

#### flatq的配置与启动

进入flatq的安装目录：/var/myproject/flatq/bin

#cd /var/myproject/flatq/bin

##### 按实际环境配置flatq

#vi mq.conf

[cmn]

#队列对应的根目录

home=/usr/home/mq/

#运行类别:master:单机版的主mq, slave:单机版的从mq, zk:分布式版

server\_type=master/slave/zk

#socket的buffer大小单位K

sock\_buf\_kbyte=8192

#最大传输chunk包大小单位K

max\_chunk\_kbyte=256

#监控host:port

monitor=172.16.42.65:9900

#同步连接数目

sync\_conn\_num=10

#该实例队列的标示ID

mq\_id=mq1

#单机版时队列topic配置文件路径

topic=/usr/home/mq/conf/topic.cnf

#是否打开debug:yes:no

debug=yes

[binlog]

#队列binlog的存放路径

path=/user/home/mq/data/binlog

#binlog文件的前缀

file\_prefix=binlog

#binlog文件的最大大小单位M

file\_max\_mbyte=1024

#binlog文件的最大数目

max\_file\_num=24

#接收多少条消息刷新磁盘

flush\_log\_num=1000

#多少秒刷新一次磁盘

flush\_log\_second=1

#是否merge

is\_merge=yes

#merge的大小

merge\_size=256

#binlog文件保存天数

save\_file\_day=5

[outer\_dispatch]

#外部分发监听端口

listen=127.0.0.1:9903

#单机版时保存source目录文件

source\_path=/data2/mq\_data/data/inner\_source

#source刷新数目

source\_flush\_num=1

#多少秒刷新次source信息

source\_flush\_second =10

[inner\_dispatch]

#内部分发监听端口

listen=127.0.0.1:9904

[recv]

#消息接收端口

listen=127.0.0.1:9901

[master]

#单机版且为slave模式时，master-mq的host:port

listen=127.0.0.1:9903

#是否支持压缩

zip=yes

[zk]

#分布式模式时对应zk的ip:port

server=ip:port;ip:port

#zk的认证信息

auth=

#zk的根目录

root=/flatq

#该实例所属组名

group=group1

##### flatq的启动与停止

在/var/myproject/flatq /目录下

**启动：**

#./flatq -f ./mq.conf >/dev/null 2>&1 &

##############运行日志输出在bin的log目录下

**停止：**

#./flatq --stop

#### product\_proxy(写代理)的配置与启动

进入product\_proxy的安装目录：/var/myproject/flatq/product\_proxy

#cd /var/myproject/flatq/product\_proxy

##### 按实际环境配置product\_proxy

#vi product\_proxy.conf

[cmn]

#product\_proxy运行的home目录，为product\_proxy所在的目录

home=/root/smothmq/src/product\_proxy

#socket-buffer的大小

sock\_buf\_kbyte=8192

#监听接收消息host:port

listen=127.0.0.1:9990

#发送消息给mq的超时时间

send\_timeout=3

#最大发送队列消息数

max\_queue\_num=1000

[zk]

#zk的hosrt:port

server=127.0.0.1:2181

#zk认证信息

auth=

#zk的根目录(flatq配置相同)

root=/flatq

##### product\_proxy的启动与停止

在/var/myproject/flatq/product\_proxy的目录下

**启动：**

#./flatq\_proproxy -f ./product\_proxy.conf >/dev/null 2>&1 &

##############运行日志输出在bin的log目录下

**停止：**

#./flatq\_proproxy --stop

#### subscribe\_proxy (读代理)的配置与启动

进入subscribe\_proxy的安装目录：/var/myproject/flatq/ subscribe\_proxy

#cd /var/myproject/flatq/ subscribe\_proxy

##### 按实际环境配置subscribe\_proxy

#vi subscribe\_proxy.conf

[cmn]

#subscribe\_proxy运行的home目录，为subscribe\_proxy所在目录

home=/root/smothmq/src/subscribe\_proxy

#socket-buffer大小

sock\_buf\_kbyte=8192

#同步链接数目

sync\_conn\_num=3

#监听端口

listen=127.0.0.1:9930

#最大chunk包大小

max\_chunk\_kbyte=2048

[zk]

#zk-server的host:port信息

server=127.0.0.1:2181

#zk认证信息

auth=

#zk根目录

root=/flatq

##### subscribe\_proxy的启动与停止

在/var/myproject/flatq/subscribe\_proxy的目录下

**启动：**

# ./flatq\_subproxy -f ./subscribe\_proxy.cnf >/dev/null 2>&1 &

##############运行日志输出在bin的log目录下

**停止：**

# ./flatq\_subproxy --stop

## 测试与接口

### 发送消息

切换到test目录

#cd test

根据环境修改send\_data.py脚本

# vim send\_data.py

修改host/send\_info为对应product\_proxy的listen host和port

Host = ‘127.0.0.1’

Send\_info = {“port”:9990, “zip”:False}

运行发送脚本

#python send\_data.py

注:刚启动mq时，需要到zk中添加topic才可发送消息，否则非法topic消息拒绝接收。

连上zk创建topic: create /flatq/group1/topic/topic1 0

### 同步消息

Cd到test目录

#cd test

根据环境修改sync\_data.py脚本

#vim sync\_data.py

修改host/sync\_info为subscribe\_proxy的listen host:port

Host=”127.0.0.1”

Sync\_info = {“port”:9930}

修改从哪个组接收哪个topic消息

pack = poco.pack\_sync\_report(0, 0, 1, "group1", "topic1", "source1", True)

启动同步消息脚本

#python sync\_data.py

### 接口

消息接口文档参见doc/interface.xml文件。

Python的接口参见interface/目录下的send\_data.py, sync\_data.py