dongfuye

非阻塞/异步(epoll) openssl

前段时间在自己的异步网络框架handy中添加openssl的支持，当时在网络上搜索了半天也没有找到很好的例子，后来自己慢慢的摸索，耗费不少时间，终于搞定。因此把相关的资料整理一下，并给出简单的例子，让后学者可以少费些力气。

同步的openssl调用网上已经有许多的例子，这里就不再详细介绍，大家也可以直接读源代码：

同步客户端：https://github.com/yedf/openssl-example/blob/master/sync-ssl-cli.cc
该例子连接www.openssl.com:443，发送一个Http请求，并打印结果中的前256个字符

同步服务器端：https://github.com/yedf/openssl-example/blob/master/sync-ssl-svr.cc
该例子监听本地的443端口，并返回一个简单http响应

下面详细介绍非阻塞调用

1. 初始化SSL库

```
SSL_load_error_strings ();
SSL_library_init ();
sslContext = SSL_CTX_new (SSLv23_method ());

//server端需要初始化证书与私钥
string cert = "server.pem", key = "server.pem";
r = SSL_CTX_use_certificate_file(g_sslCtx, cert.c_str(), SSL_FILETYPE_PEM);
r = SSL_CTX_use_PrivateKey_file(g_sslCtx, key.c_str(), SSL_FILETYPE_PEM);
r = SSL_CTX_check_private_key(g_sslCtx);
```

2. 非阻塞方式建立tcp连接（网上有很多epoll相关例子）

3. 使用已建立连接的socket初始化ssl

```
ch->ssl_ = SSL_new (g_sslCtx);
int r = SSL_set_fd(ch->ssl_, ch->fd_);
服务器端 SSL_set_accept_state(ch->ssl_);
客户端 SSL_set_connect_state(ch->ssl_);
```

4. epoll_wait后，如果SSL相关的socket有读写事件需要处理则进行SSL握手，直到握手完成

```
int r = SSL_do_handshake(ch->ssl_);
if (r == 1) { // 若返回值为1，则SSL握手已完成
    ch->sslConnected_ = true;
    return;
}
int err = SSL_get_error(ch->ssl_, r);
if (err == SSL_ERROR_WANT_WRITE) { //SSL需要在非阻塞socket可写时写入数据
    ch->events_ |= EPOLLOUT;
    ch->events_ &= ~EPOLLIN;
} else if (err == SSL_ERROR_WANT_READ) { //SSL需要在非阻塞socket可读时读入数据
    ch->events_ |= EPOLLIN; //等待socket可读
    ch->events_ &= ~EPOLLOUT; //暂时不关注socket可写状态
} else { //错误
    ERR_print_errors(errBio);
}
```

导航
博客园
首页
新随笔
联系
订阅
管理

公告

昵称： dongfuye
园龄： 7年4个月
粉丝： 16
关注： 0
+加关注

< 2021年12月 >

日	一	二	三	四	五	六
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

统计

随笔 - 11
文章 - 0
评论 - 8
阅读 - 45571

搜索

找找看
谷歌搜索

常用链接

我的随笔
我的评论

5. 握手完成后，进行SSL数据的读写

```
SSL_write(con->sslHandle, text, len);
SSL_read(con->sslHandle, buf, sizeof buf);
```

详细可运行的例子参看

https://github.com/yedf/openssl-example/blob/master/async-ssl-svr.cc

https://github.com/yedf/openssl-example/blob/master/async-ssl-cli.cc

handy已经对openssl进行了封装，并且给出了例子，详见

https://github.com/yedf/handy-ssl

标签: c++, 服务器, openssl, epoll

dongfuye
关注 - 0
粉丝 - 16
+加关注

« 上一篇： C++ 高性能无锁日志系统
» 下一篇： 阿里云 SDK python3支持

posted on 2014-11-25 15:04 dongfuye 阅读(12215) 评论(0) 编辑 收藏 举报

我的参与
最新评论
我的标签

我的标签

网络(5)
c++(4)
c++11(3)
epoll(3)
服务器(3)
微服务(2)
kqueue(2)
高性能(2)
linux(2)
分布式事务(1)
更多

随笔档案

2021年7月(1)
2017年1月(1)
2016年3月(1)
2015年11月(1)
2015年8月(2)
2015年6月(3)
2014年11月(1)
2014年8月(1)

阅读排行榜

1. 非阻塞/异步(epoll) openssl(12215)
2. 单机千万并发连接实战(修订版)(6302)
3. 千万并发连接实战(6155)
4. C++11网络编程(5100)
5. C++ 高性能无锁日志系统(3437)

评论排行榜

1. 千万并发连接实战(3)
2. kqueue例子(2)
3. C++ 高性能无锁日志系统(2)
4. 阿里云 SDK python3支持(1)

推荐排行榜

1. 千万并发连接实战(12)
2. 单机千万并发连接实战(修订版)(11)
3. kqueue例子(3)
4. C++ 高性能无锁日志系统(3)
5. 非阻塞/异步(epoll) openssl(1)

最新评论

1. Re:阿里云 SDK python3支持
您好，您这个再python3现在跑不了啊
--病毒尖er

2. Re:C++ 高性能无锁日志系统
你的思路是指每天定时给程序发送一个信号，让程序轮替日志，这是许多应用的做法，我印象中的mongo是这样。思路没有问题，细节有点出入，在write时，接收到信号，不同的操作系统会由不同的表现：. wri...
--dongfuye

3. Re:C++ 高性能无锁日志系统
您好！咨询一个问题，如果write操作的时候，接收到信号，在信号处理函数里面执行dup2，把write对应的fd修改为新打开的文件的fd，那么信号处理返回后，原来的write操作继续执行，会不会出问题...
--WONDERFUL_cnblogs

4. Re:千万并发连接实战
博主，杠杠的
--lijihong0723

5. Re:千万并发连接实战

mark

--hhhhj