[thrift开发踩坑全纪录](http://blog.csdn.net/bestwolf1983/article/details/52074914)

标签： [thrift](http://www.csdn.net/tag/thrift)

2016-07-30 20:50 580人阅读 [评论](http://blog.csdn.net/bestwolf1983/article/details/52074914#comments)(0) [收藏](javascript:void(0);) [举报](http://blog.csdn.net/bestwolf1983/article/details/52074914#report)

http://static.blog.csdn.net/images/category_icon.jpg 分类：

C++开发（13） http://static.blog.csdn.net/images/arrow_triangle%20_down.jpg

版权声明：本文为博主原创文章，未经博主允许不得转载。

使用thrift搭建的系统已经稳定运行了一段时间了，该系统是公司的核心流式系统，高峰时QPS在40万。作为目前最流行的RPC框架，thrift不仅提供了通信协议，同时提供了网络框架，解脱了程序员的生产力。thrift也是阿帕奇[**Hadoop**](http://lib.csdn.net/base/hadoop)系列的RPC实现工具。本文主要聚焦在搭建C++实现的thrift系统中，遇到的各种问题。

但是thrift在隐藏一些底层细节的同时，也给应用层带来了一些不确定性，这些不确定和误解，导致一些异常事件的发生。总结如下：

1. thrift compile在生成文件的时候，生成PHP文件时，同一命名空间下，不同的thrift文件生成Types.php，会导致文件覆盖（cpp和Python没有此问题）
2. 服务端在采用TBufferedTransport时，该对象实际会自己缓存read/write buffer，当catch到异常时，仅仅close对应的socket，buffer并不会清空。此时如果再次open socket发送数据，会导致buffer数据混乱，transport出现异常。解决的方法是把指向该TBufferedTransport的shared\_ptr.reset()。
3. 服务端和客户端采用的协议必须一样。服务端用TBufferedTransport，客户端不能用TFrameworkTransport，否则会异常。客户端报can't write xxx bytes的异常
4. 有时候会设置服务端和客户端的超时，这时候如果另外一方在接/发数据的时候，未能在timeout时间内完成数据传输，则会由于另外一方的强制断开而报异常。异常的内容是：broken pipe
5. 如果thrift服务经常catch到异常，且异常的内容每次都不同，如broken pipe, no more data to read等，很有可能是在客户端，多个线程共用一个socket，却没有加锁导致的Heisenbug
6. 服务端hander中，用户实现的服务接口，不能有block的代码，如sleep()，lock.wait()，否则会导致工作线程被阻塞。这样引起的后果是：1，客户端请求线程被卡死在socket::read()；2，服务端fd泄露
7. thrfit的client不能在多个线程间不加同步的复用，因为client不是线程安全的。