刘旭东
软件复用
April 10, 2016

软件复用讨论课

如何能够保持用户登录后始终在线

在这次的项目中，为了能够实现保持用户登录后始终在线的这个特性，我们采用了ActiveMQ的框架。ActiveMQ框架自身带有保持连接的机制，以及断开重连的特性。

ActiveMQ的心跳保持连接机制：

为了更好的维护TCP链路的使用，ActiveMQ采用了心跳机制作为判断双方链路的健康情况。ActiveMQ使用的是双向心跳，也就是ActiveMQ的Broker和Client双方都进行相互心跳，但不管是Broker或Client心跳的具体处理情况是完全一样的，都在InactivityMonitor类中实现。

心跳会产生两个线程“InactivityMonitor ReadCheck”和“InactivityMonitor WriteCheck”，它们都是Timer类型，都会隔一段固定时间被调用一次。ReadCheck线程主要调用的方法是readCheck()，当在等待时间内，有消息接收到，则该方法会返回true。WriteCheck线程主要调用的方法是writeCheck()，其实当WriteCheck线程休眠时，有任何数据发送成功，则该线程被唤醒后，不用通过TCP向对方真的发送心跳消息，这样可以从一定程度上减少网络传输的数据量。

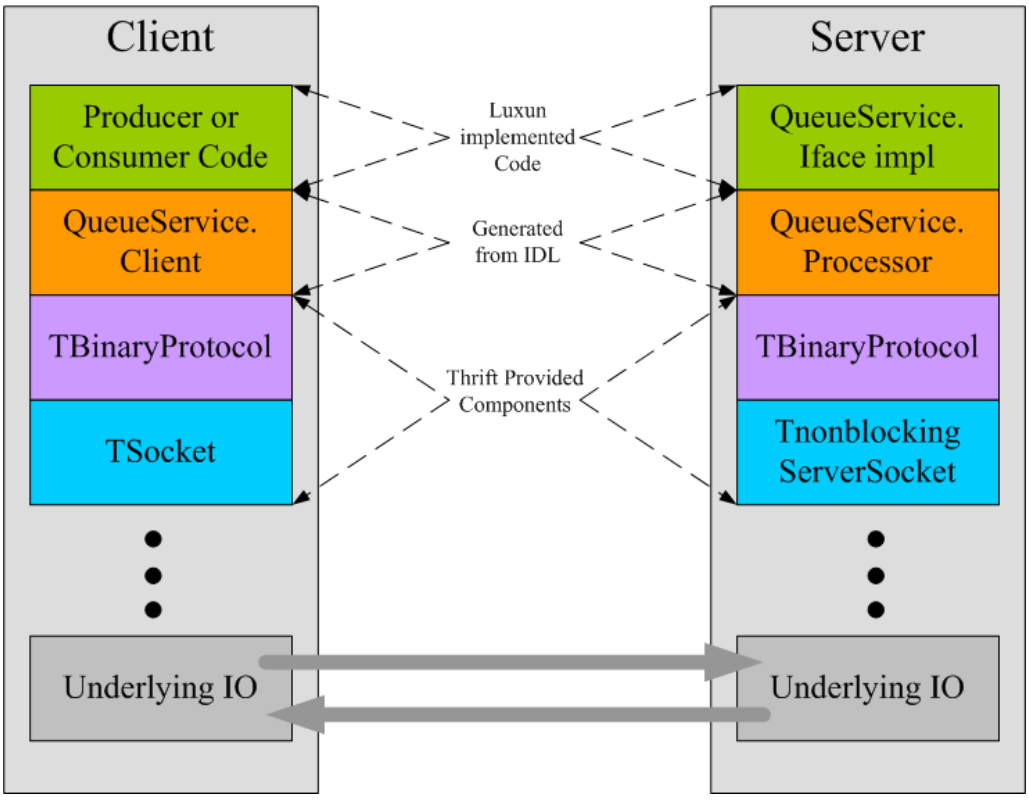
如果你想时时监听整个连接的状态，可以添加一个ActiveMQ的消息传输监听，实现 ActiveMQ的TransportListener接口。该接口是有onCommand（），onException（），transportResumed () 等监听方法。这样你就可以通过ActiveMQ实时了解客户端的服务器端的状态以及原因了。ActiveMQ还支持自定义设置心跳的时间等等一些自定义化的设置，可以使我们能够让它更符合我们的实际需求。

ActiveMQ的断线重新连接机制：

断线重连机制是ActiveMQ的高可用性具体体现之一。其中有Failover Transport和Discovery transport 两种transport的方式，Failover Transpor是一种重新连接的机制，它工作于其它transport的上层，用于建立可靠的传输。具体就是使用failover方式，使得连接断开之后，可以不断的重试连接到一个或多个brokerURL。在failover:(tcp://127.0.0.1:61616)，这里可以使用多个url。默认情况下，如果client与broker直接的connection断开，则client会新起一个线程，不断的从url参数中获取一个url来重试连接。而 Discovery transport是可靠的tranport。它使用Discovery transport来定位用来连接的URI列表。

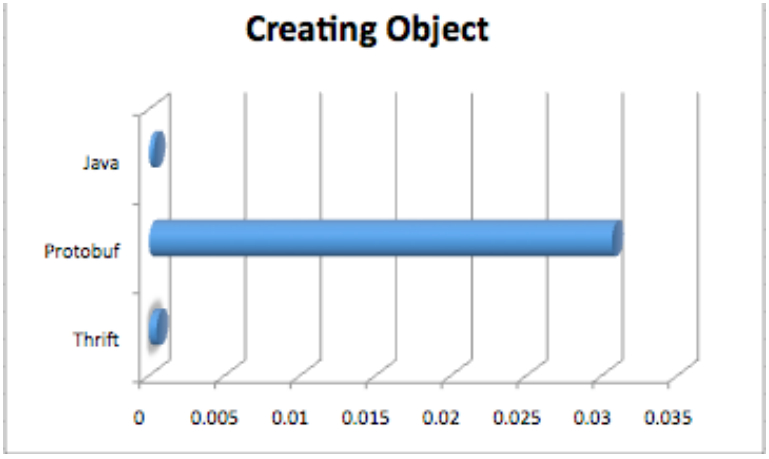
消息的压缩：

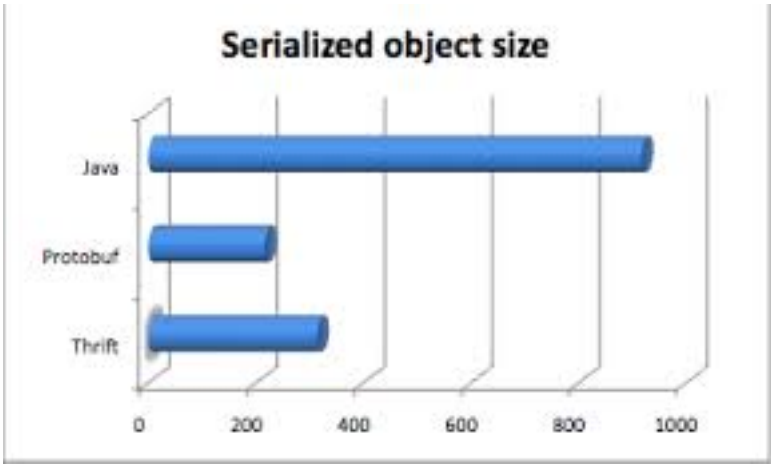
我们可以使用Thrift来对消息进行压缩。Thrift是跨语言的RPC框架，而且是基于C/S模式的一个框架，现在是一个Apache的顶级项目。其传输数据采用二进制格式，相对 XML 和 JSON 体积更小，对于高并发、大数据量和多语言的环境更有优势。Thrift通过一个中间语言—IDL接口定义语言，来定义RPC的接口和数据类型。使用Thrift的代码生成工具(thrift-0.9.1.exe编译器)读取IDL文件，生成不同语言的服务端与客户端代码,并由生成的代码负责RPC协议层和传输层的实现。目前支持语言C++,Java, Python, PHP, Ruby, Erlang, Perl, Haskell, C#, Cocoa, Smalltalk.



保持连接

采用Thrift框架之后，对消息的压缩情况如下：





可以看到，thrift可以有效地降低一个消息的大小，而且并没有牺牲创建消息的速度。Thrift提供了一整套的使用框架，但是由于缺乏一些API文档，可能使用起来学习成本会有一些高，但是如果需要使用压缩的话，采用Thrift的方法能够大幅度降低消息的大小，从而使得传输更有效率。

低宽带，网络不稳定情况下：

通过设定每个请求的time out时间，当请求超过这个设定值后，提示用户网络请求超时，将刚刚的请求撤销，保存在本地。用户可以在网络状况变好之后，选择重新发送或者撤销重新发送新的内容。