

软件复用解决方案

1352863 张嘉琦

条件：登录后始终在线，考虑低带宽、不稳定网络。

主要考虑中间件：activemq

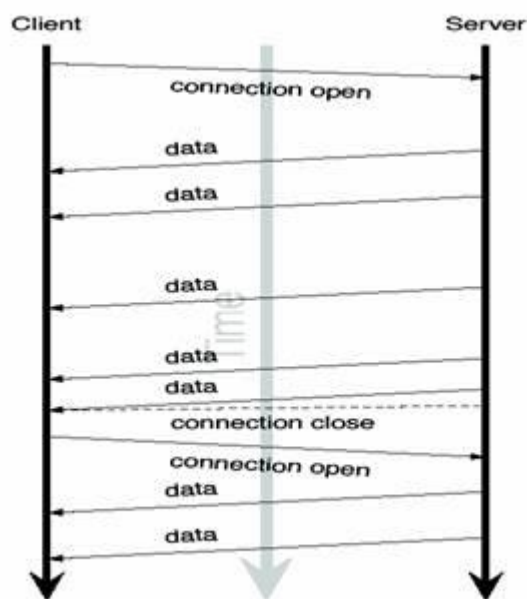
一.

长连接心跳机制：

定义：

维护任何一个长连接都需要心跳机制，客户端发送一个心跳给服务器，服务器给客户端一个心跳应答，这样就形成客户端服务器的一次完整的握手，这个握手是让双方都知道他们之间的连接是没有断开，客户端是在线的。如果超过一个时间的阈值，客户端没有收到服务器的应答，或者服务器没有收到客户端的心跳，那么对客户端来说则断开与服务器的连接重新建立一个连接，对服务器来说只要断开这个连接即可。

长连接工作原理



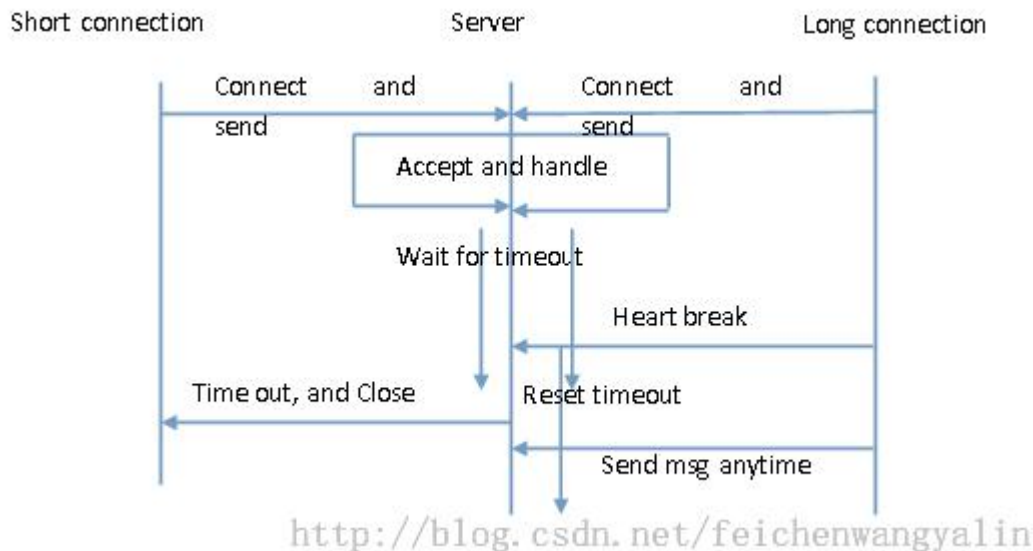
解决方案：

从上图可以看出每次数据传送不会关闭连接，连接只会在通信出现错误时，或是连接重建时关闭（一些防火墙常被设置为丢弃过长的连接，服务器端可以设置一个超时时间，超时后通知客户端重新建立连接，并关闭原来的连接）。

Socket：

socket 实际上是对 TCP/IP 进行的封装，我们可以使用 socket 套接字通过 socket 来传输。首先我们需要明白的一个概念就是通道，简单地说通道就是两个对端可以随时传输数据的信道。我们常说的所谓建立 socket 连接，也就是建立了客户端与服务器端的通道。Java 中的 socket 本身就是支持长连接的，可是假如服务器不加以任何限制地直接和客户端使用长连接，服务器需要承受极大的压力，所以一般的 socket 服务器端都是会设定超时时间的，也就是

timeout, 如果超过 timeout 服务器没有接收到任何数据, 那么该服务器就会关闭该连接, 从而使得服务器资源得到有效地使用。为了实现长连接, 客户端在 timeout 时间前一秒 (或者更短的时间) 发送一条激活数据来使服务器端重新计时, 如此重复就能保证服务器一直不能进入 timeout 时间, 从而一直保持连接, 这就是长连接的实现原理



由上图可见, 是否是长连接完全取决于客户端是否会在 timeout 时间发送心跳消息, 因此长短连接是和客户端相关的, 服务器端没有任何区别 (只不过服务器端需要设定 timeout 而已)。

http 的长连接:

HTTP 也可以建立长连接的, 使用 `Connection:keep-alive`, HTTP 1.1 默认进行持久连接。HTTP1.1 和 HTTP1.0 相比较而言, 最大的区别就是增加了持久连接支持(貌似最新的 http1.0 可以显示的指定 `keep-alive`), 但还是无状态的, 或者说是不可信任的

Activemq 断线重连:

为了完成长连接, activemq 的断线重连机制是我们选择它的一个主要因素。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`。

例如: `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。这个机制对于在容器中使用的 connection 木有问题。我们在进行项目开发调试的时候, 也发现在断线之后, client 尝试重连。由于多次重连, 还给我们的调试造成了一点困扰。

二. 消息不遗漏:

当 client 发送消息时, 服务器收到消息就会发回一个回执, 返回是否发送成功, 这样就完成了一次握手, 保证了消息的不遗漏。

当网络不稳定的时候, 只要保证收到了发送成功的回复, 那么基本上就能避免消息的遗漏。但是, 对于在发送过程中, 由于网络或者软件的原因已经造成了消息的遗失, 我们想到两种解决方法: 1. 提醒消息发送方进行重新发送。 2. 对 activeMQ 使用消息持久化的技术, 将消息存到数据库中, 以 Mysql 为例: 当消息发送至 ActiveMQ 时, 数据就被持久化到 mysql 了, 如果消息被消费, 数据会自动被删除, down 机后重启没影响, 有一点不好的是, 这个有

点拖数据库，我在本地的 mysql，一开启 ActiveMQ，数据库就会变得很慢，不过这个只是在本地的机子上，想必实际应用时应该好很多。

三. 消息不重复

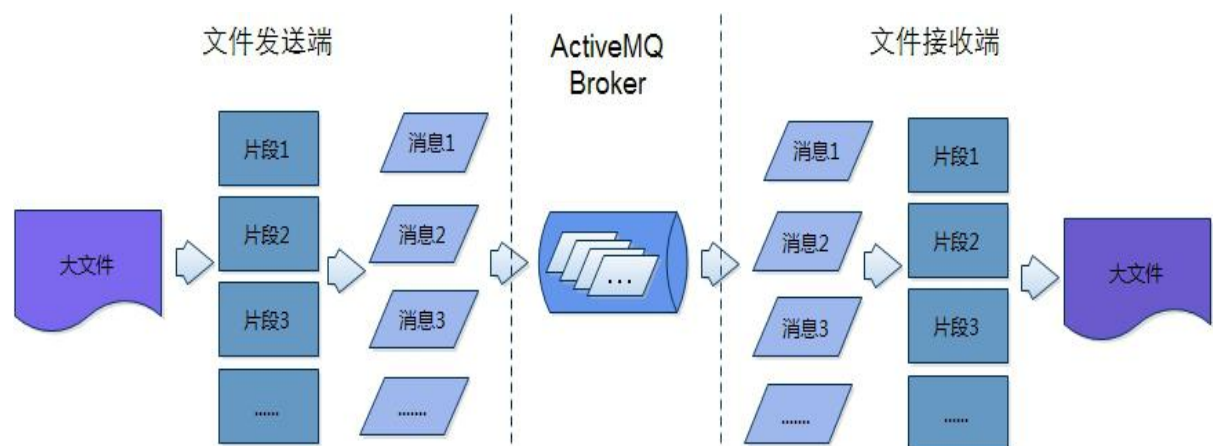
当使用消息机制的时候，为了避免消息遗漏都会设置一些重发机制，而重发的时候，如果之前发送失败，或者由于网络等等原因没有发送成功的信息有恢复正常了，就会重复发送。对此的解决方案有：

业务端增加一张表存放，消息是否执行。每次业务事务 commit 后，告知 server 端已经处理该消息。即使 server 因为接受 timeout 重发，导致多个业务机器收到消息，也不会导致重复处理。

四. 消息压缩

activemq 发送文件：

1. 使用 connection.createOutputStream ()



2. 使用文件转发

- 1) 首先使用共享或 ftp、dfs 等方式，先把文件发送到一个大家都可以拿到的地方。
- 2) 通过 producer.send 将文件的 URL 信息发送到 client
- 3) Client 通过 url 获取文件。

