文章
Louis Lu · 九月 21 阅读大约需 2 分钟

Production 中调用 web service 组件时有关超时参数的设置

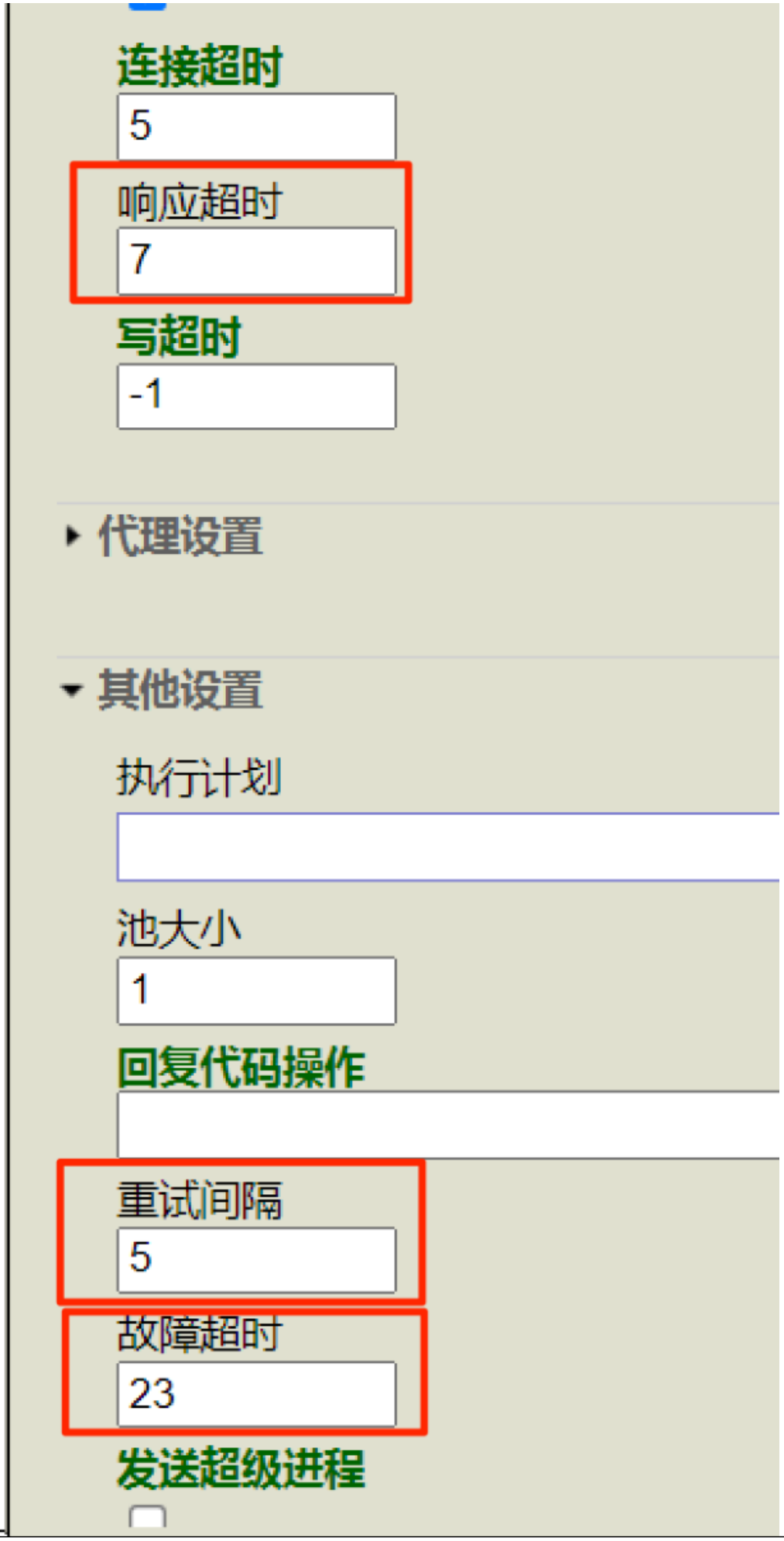
我在这里和大家分享下在 Interoperability 的接口开发中，调用Web Service接口的几个超时参数的设置经验。

赶时间的同学可以直接拉到文章最下面看结论就好。

1.实验过程

首先我设计了一个Web service的服务器端，强制在接收到请求后 8s 返回结果。

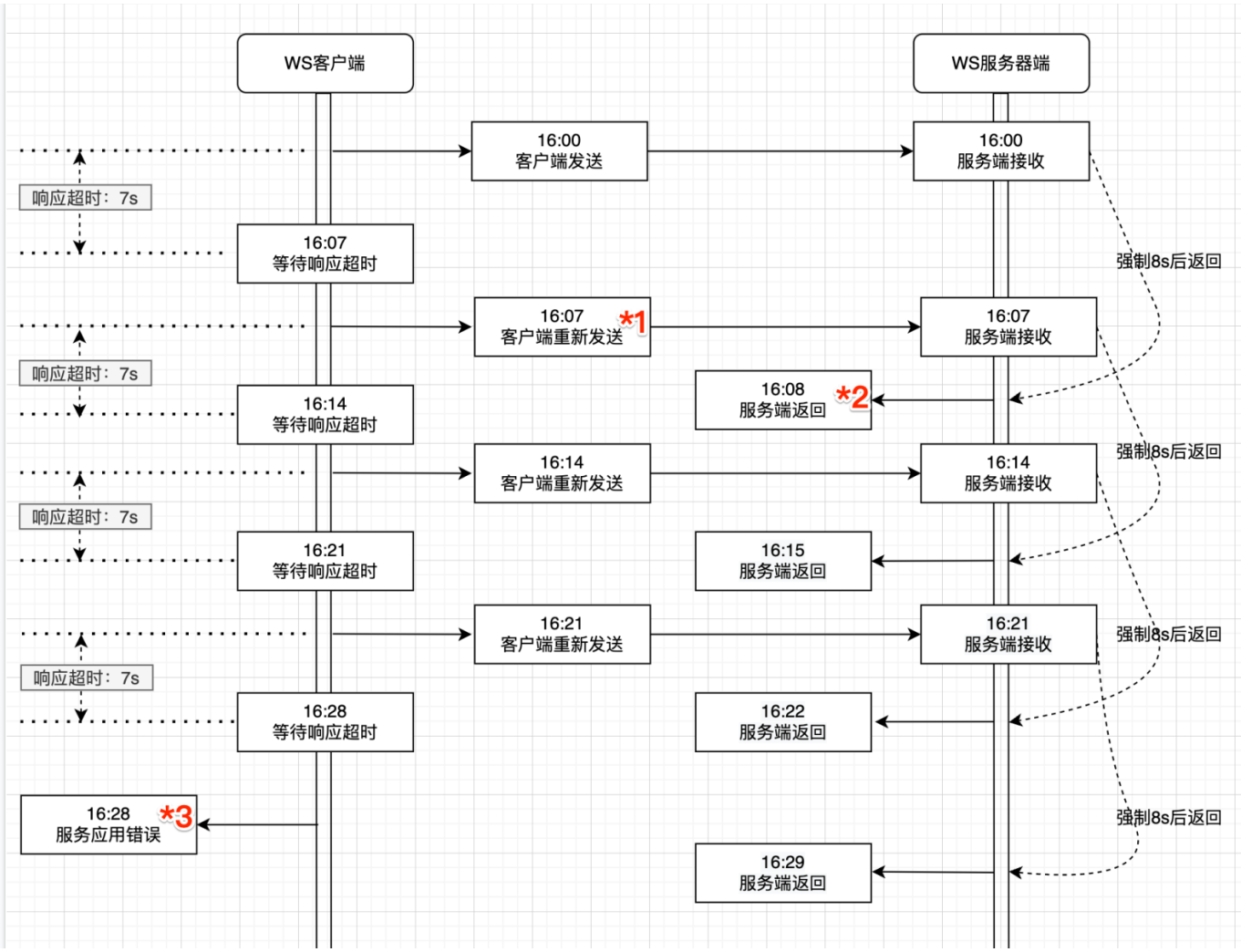
在客户端我设置了**响应超时7s, 重试间隔5s, 故障超时23s**，如图：


连接超时
5
响应超时
7
写超时
-1
代理设置
其他设置
执行计划
池大小
1
回复代码操作
重试间隔
5
故障超时
23
发送超级进程

在客户端、服务器端均设置了SOAP Log 记录接收和发送的内容

```
set ^ISCSOAP("Log")="io"
set ^ISCSOAP("LogFile")="c:\temp\soapClient.txt"
```

经过整理日志我画了下面的数据流图



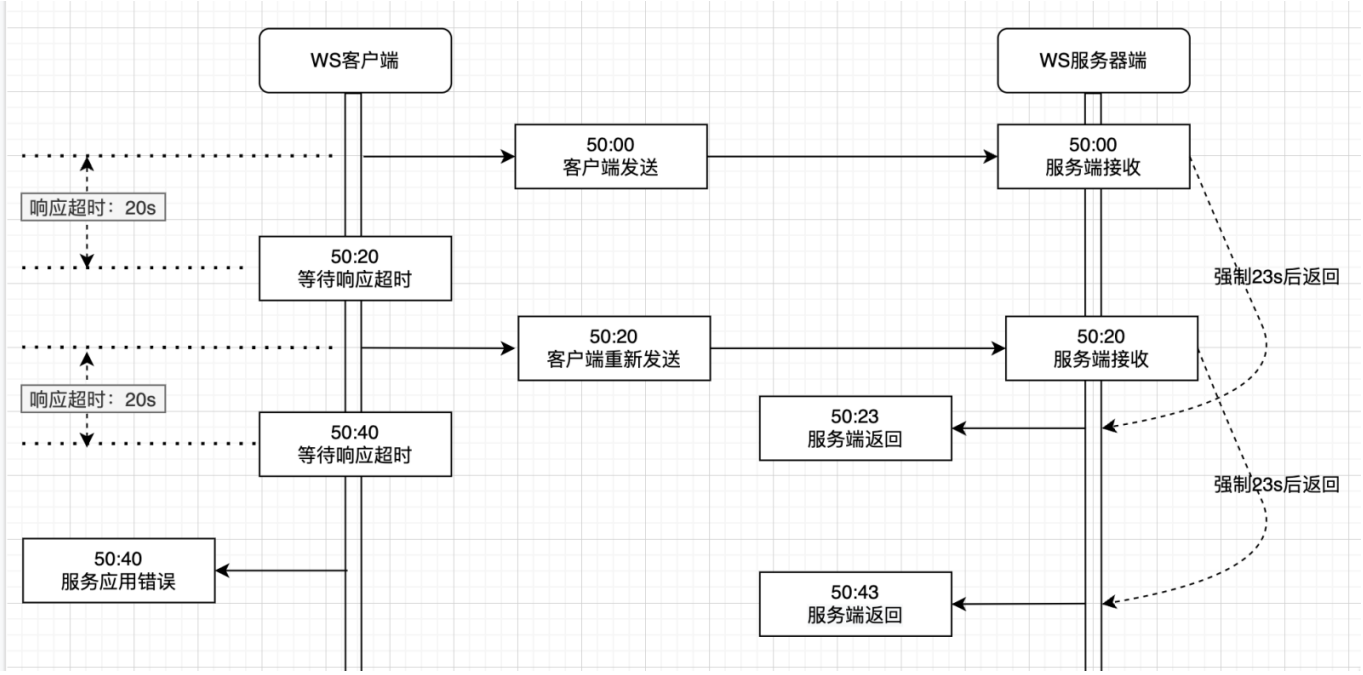
2. 得出下面**结论**：

- *1：重试间隔虽然设置为5s，但是仅当有响应超时错误后才会被触发
- *2：虽然服务器端有返回值，但是已经超过了设定的响应超时(7s)时间，则返回值不会被客户端接收
- *3：故障超时虽然设置为23s，但是仅当有响应超时错误后才会被触发

3. 结论再验证

为了验证上面结论我修改了Web service的服务器端代码，强制在接收到请求后 23s 返回结果，

并且设置**响应超时20s, 重试间隔6s,**
故障超时25s，得出上面同样的结论：自动重发或者故障超时错误的触发条件都是收到响应超时错误。



所以我们要特别注意设置**响应超时**这个参数的值：

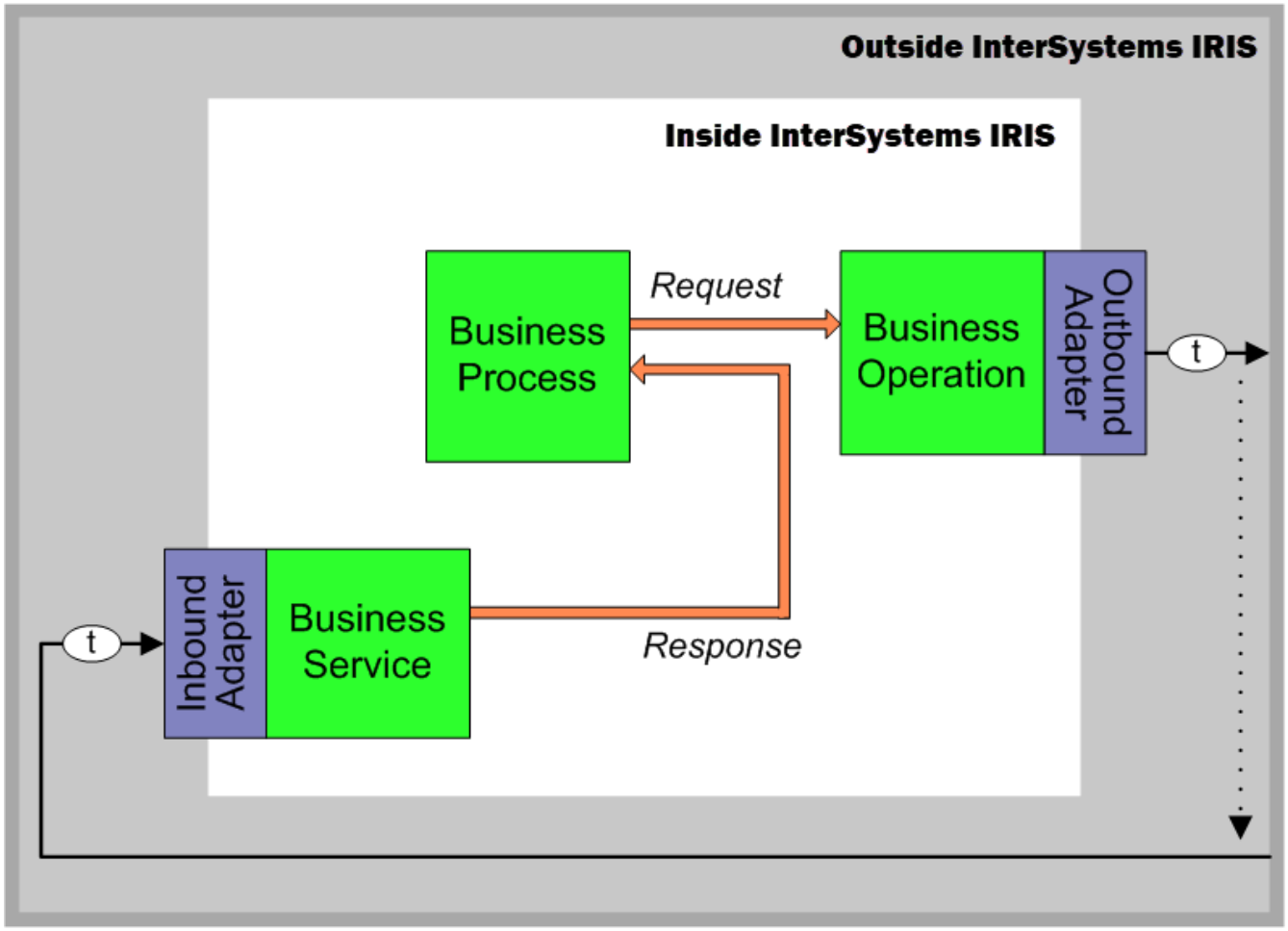
1. 如果设置的值过短，特别是小于服务器端返回值的时间，则客户端一定接收不到返回值
2. 如果设置的值过长，那么该进程会一直处于等待结果的状态，其他的请求会被放入到队列中，从而减低处理效率，系统资源造成浪费。

4. 进一步思考

如果熟悉 Production开发的人一定知道，只能在Services
或者Processes中设定消息的发送是以同步或者异步的方式，那么对于Operations我们怎么处理呢？

特别是像遇到这篇文章中的情况，接入的三方系统可能需要比较长的时间才能返回结果，如果 Production 每次都需要一个进程等待结果返回，只有收到返回后才能把该进程释放出来处理另一个消息请求，则必然造成系统资源的浪费。

这种情况我推荐使用消息传递的第三种方式 Deferred
sending，也就是通过Servies接收到的内容构建原Operations的返回消息，如下图：



有关使用deferred sending的例子我会在之后的文章中分享。

#InterSystems IRIS

源URL:https://cn.community.intersystems.com/post/production-%E4%B8%AD%E8%B0%83%E7%94%A8-web-service-%E7%BB%84%E4%BB%B6%E6%97%B6%E6%9C%89%E5%85%B3%E8%B6%85%E6%97%B6%E5%8F%82%E6%95%B0%E7%9A%84%E8%AE%BE%E7%BD%AE