Dubbo探研及浅析

我们最好从dubbo的官方高清大图入手看整体架构



一、从垂直纵向看结构的层次

1.网络服务层

服务层用作传输对象的实体需要继承serializable接口

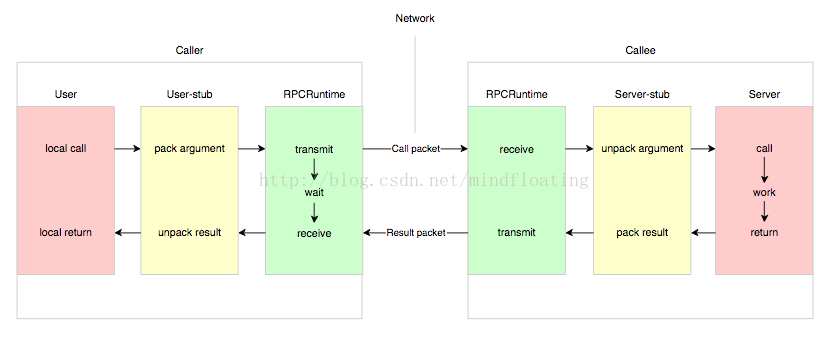
2.基于RPC的dubbo网络协议层

实现 RPC 的程序包括 5 个部分：User、 User-stub、 RPCRuntime、Server-stub、Server

（1）config服务配置

（2）proxy服务代理

（3）register服务注册



（4）cluster服务调用方

（5）monitor服务监控

（6）protocol服务中心

3.网络传输层

在serialize层根据来源的不同，进行序列化和反序列化。

二、从水平横向看模块组成

从水平方向把图平均分为两半，右半边是服务提供方，左半边则是服务调用方。

我们可以看出整个流程是：

从右上角的服务提供方开始-->（服务配置-->服务代理-->服务注册-->服务监控-->服务中心)；

然后再从：

左上方的服务调用方需要服务开始-->（消费配置-->消费代理-->消费注册-->服务监控-->服务中心)；最后从：

左上方的服务调用方进行消费开始-->从服务配置获取服务接口-->获取服务代理-->服务过滤与服务监控-->网络传输层-->服务过滤-->服务代理-->服务提供方实现服务.

1.服务提供方

首先创建服务。

*<!-- 服务提供方信息，用于辨别依赖关系 -->*<dubbo:application name="sv\_provider" owner="zhangzh" />

*<!-- 使用zookeeper注册中心暴露服务地址->即zookeeper的所在服务器ip地址和端口号 -->*<dubbo:registry id="honeycatRegistry" address="zookeeper://127.0.0.1:2181"/>  
*<!-- 用dubbo协议在20880端口暴露服务 -->*<dubbo:protocol name="dubbo" port="20880"/>  
*<!-- 用hessian协议在20887端口暴露服务 -->*<dubbo:protocol name="hessian" port="20887"/>  
*<!-- 提供方参数version\_old -->*<dubbo:provider delay="-1" timeout="10000" retries="3" >  
 <dubbo:parameter key="catId" value="honey1"/>  
 <dubbo:parameter key="package" value="honeycat\_OPENAPI"/>  
</dubbo:provider>

然后暴露服务。

*<!-- teacher-info使用dubbo协议暴露服务 -->*<bean id="teacherClient" class="honeycat.lay.dubbo.teacher.clientImpl.TeacherClientImpl">  
</bean>  
<dubbo:service interface="honeycat.lay.dubbo.teacher.client.TeacherClient" owner="zhangzh"  
 ref="teacherClient" version="0.1" protocol="dubbo" delay="-1" registry="honeycatRegistry"/>

2.服务调用方

调用dubbo服务进行消费。

<dubbo:reference id="teacherClient"  
 registry="honeycatRegistry" owner="zhangzh"  
 interface="honeycat.lay.dubbo.teacher.client.TeacherClient"  
 version="0.1" protocol="dubbo" timeout="100000" init="true"  
 check="false"/>

基于dubbo的文件上传

一种是用servlet方式，此方式需要在web.xml配置

个人觉得servlet 3.0的注解可以代替在web.xml内的配置，应该可用，但是Part与那版本就是获取前端的文件；又怎么改为从controller获取，显然协议或方式是需要改变的。

我们这里使用hessian协议进行传输(当然也可以转换成byte[]继续使用dubbo协议) <http://blog.csdn.net/qq315737546/article/details/52792037>

# 参考：

[**dubbo 配置文件详解**](http://www.cnblogs.com/linjiqin/p/5859153.html)<https://www.cnblogs.com/linjiqin/p/5859153.html>

<http://blog.csdn.net/hejingyuan6/article/details/47403299> dubbo实战

<http://blog.csdn.net/accp_fangjian/article/details/51658292> dubbo+zookeeper+ssm

<http://doc.okbase.net/congcong68/archive/112508.html>（dubbo+zookeeper介绍、搭建、使用）

<http://blog.csdn.net/shiyus1314/article/details/77500780> **使用dubbo和zookeeper的一些理解**

dubbo一些详解的地址：<http://shiyanjun.cn/archives/325.html>

**Dubbo中消费者初始化的过程解析** <https://www.jianshu.com/p/76c14af021e2>

**Dubbo+zookeeper的引入及使用** <http://blog.csdn.net/lijn_huo/article/details/51690759>

**Dubbo通信模型** <http://blog.csdn.net/qq418517226/article/details/51906357>

**Dubbo架构与底层实现** <https://yq.aliyun.com/articles/38380>

**Dubbo架构设计详解** <http://shiyanjun.cn/archives/325.html>

# 问题与解决：

dubbo间歇性超时问题解决<http://blog.csdn.net/csujiangyu/article/details/51984600>

Caused by: org.xml.sax.SAXParseException; lineNumber: 6; columnNumber: 61; cvc-complex-type.2.4.c: 通配符的匹配很全面, 但无法找到元素 'dubbo:application' 的声明。

Jar冲突

<https://www.cnblogs.com/hfultrastrong/p/7876345.html>

Xml文件头部

<http://blog.csdn.net/eacter/article/details/44624505>

Exception in thread "main" java.lang.NoClassDefFoundError: org/I0Itec/zkclient/IZkStateListener

缺jar（zkclient）

<https://www.cnblogs.com/lujion/p/6193720.html>

打开tomcat的登录页面账户和密码

<http://blog.csdn.net/sweblish/article/details/6934327>

登录tomcat 出现access denied

<http://blog.csdn.net/tajun77/article/details/51330876>

解决 <http://blog.csdn.net/itlsx/article/details/26251695>

Tomcat启动闪退

<https://jingyan.baidu.com/article/9f7e7ec0be98b56f281554e9.html>

# 代码启动logback日志

如何加载指定路径的Logback.xml

<http://bbs.csdn.net/topics/380180401>

<http://blog.csdn.net/coffee_hc/article/details/44424513>

代码手动加载logback日志；尝试了但是失败，1.找文件；2.loggerContext加载失败

*//读取logback.xml开启logback日志*LoggerContext lc = (LoggerContext)LoggerFactory.*getILoggerFactory*();  
JoranConfigurator configurator = **new** JoranConfigurator();  
configurator.setContext(lc);  
lc.reset();  
**try** {  
 configurator.doConfigure(**"server-client-impl/src/main/resources/config/logback.xml"**);  
} **catch** (JoranException e) {  
 **throw new** RuntimeException(**"没找到文件，请检查文件路径是否正确!"**+e.getMessage());  
}  
StatusPrinter.*printInCaseOfErrorsOrWarnings*(lc);

[**java.lang.ClassCastException: org.slf4j.impl.Log4jLoggerFactory cannot be cast to ch.qos.logback.classic.LoggerContext问题原因及解决方法**](http://www.cnblogs.com/simplelei/p/5408029.html)

<https://www.cnblogs.com/simplelei/p/5408029.html>

《SLF4J官方文档》SLF4J警告或错误信息及其含义

**java.lang.NoClassDefFoundError: org/slf4j/event/LoggingEvent // SLF4J 1.4.0版本，需要1.2.12及以上的版本**

<http://ifeve.com/slf4j-warn-error/>

[**Unable to locate Spring NamespaceHandler for XML schema namespace**](http://www.cnblogs.com/icewee/articles/3703491.html)

<https://www.cnblogs.com/icewee/articles/3703491.html> spring的命名空间错误

**使用Spring框架服务器加载xml文件的过程**

<http://blog.csdn.net/nyhyn/article/details/48423577>

[web.xml启动顺序](http://www.cnblogs.com/loveok-56/p/6433425.html)

<https://www.cnblogs.com/loveok-56/p/6433425.html>

**LogBack配置使用**

<http://blog.csdn.net/tigerd1995/article/details/50845999>

[**logback的使用和logback.xml详解**](http://www.cnblogs.com/warking/p/5710303.html)

[**https://www.cnblogs.com/warking/p/5710303.html**](https://www.cnblogs.com/warking/p/5710303.html)

**开启Dubbo框架内部的日志**

<http://blog.csdn.net/jdream314/article/details/44620767>

开启dubbo的log4j等格式日志

<http://blog.csdn.net/jdream314/article/details/44620767>