

JSF支持分组路由预研

被章耿添加, 被章耿最后更新于八月 12, 2015

需求

需求来源于分布式交易项目:

- 1.期望的效果是, 一个RPC call 里如果有pinId 参数, 需要把pinId 映射到某个机房, 然后请求对应机房的那个服务提供方。
- 2.pinID 和机房的映射关系, 在用户组那边会有个服务提供出来。

设计

在JSF / SAF中, 一组服务被标示为Group (Alias), 所以在这个场景我们需要完成的映射如下:

pinID --> 机房标示 --> SAF Group(JSF Alias);

对于pinId到机房标示的转换, 建议由业务代码来完成, 避免服务框架对其他接口的依赖;

以下设计以JSF来举例, SAF的解决方案类似, 只是相应映射规则数据存储于zookeeper上:

JSF Consumer与Alias是一对一的对应关系, 所以需要增加一个ConsumerGroup的tag, 一个ConsumerGroup 对应于多个JSF consumer, 除了alias属性可以配置多个分组外, ConsumerGroup其它配置和Consumer一样, 减少学习成本。

机房标示到JSF Alias的映射规则, 由JSF注册中心完成相应的规则维护、规则数据的推送工作;

规则举例: "room1-->Alias1;room2-->Alias2";

机房标示"room1",可以用两种方式通知给JSF服务框架:

- 1.在参数列表里面, 显式带上一个机房标示; 比如第三个参数是机房标示, 那么我们在XML配置里面配置"dstRoomParam= 2";
- 2.采用RpcContext (ThreadLocal) 机制, 隐式传参带一个机房标示, 存放在一个预先约定好的地方。例如:
RpcContext.getContext().setAttachment(".dstRoom", "yz");

在注册中心管理端配置规则"room1-->GroupA";

调用时会根据实际参数值或者ThreadLocal变量的值, 将RPC请求发给相应的Group;

实现示例:

配置和consumer一样, 区别是alias可以设置多个

```
<jsf:consumerGroup id="helloServiceGr1" interface="com.jd.testjsf.HelloService"
    alias="S1,S2" protocol="jsf" dstRoomParam="0"> <!-- 接口下所有方法默认第一个为机房参数 -->
    <jsf:method name="echoStr" dstRoomParam="0"/> <!-- 方法级 第一个参数为机房 -->
    <jsf:method name="put" dstRoomParam="1"/> <!-- 方法级 第二个参数为机房 -->
</jsf:consumerGroup>
```

ConsumerGroup生成代理类的, 会自动生成多个Consumer实例;

在业务代码请求的时候, 先走规则匹配逻辑, 定位到某个Alias, 然后通过alias对应的Consumer请求发送给相应的分组。

如果当前alias不可用:

1. 指向可用alias;
2. 抛异常??

使用实例

配置下jsf的配置（GroupRouter+ ConsumerGroup）。

```
<jsf:consumerGroup id="helloServiceGr1" interface="com.jd.testjsf.HelloService"
    alias="S1,S2" protocol="jsf">
    <jsf:method name="echoStr" dstRoomParam="0"/>
    <jsf:method name="put" dstRoomParam="1"/>
</jsf:consumerGroup>
```

业务代码里正常使用即可

```
HelloService service = (HelloService) appContext.getBean("helloServiceGr");

service.echoStr("yz"); // 调到亦庄机房
service.put(2, "hc"); // 调到黄村机房

RpcContext.getContext().setAttachment(".dstRoom", "hc");
service.echoStr("yz"); // 将会被强制调到黄村机房
```

无
