示例

**启动时检查**

**集群容错**：failover、failfaster、failsafe、failback、forking、broadcast

cluster="failsafe"

**负载均衡**：random、roundrobin、leastActive、consistentHash

**线程模型**：

<dubbo:protocol name="dubbo" dispatcher="all" threadpool="fixed" threads="100" />

dispatcher

all：所有消息都派发到线程池

direct：所有消息都不派发到线程池，全部在IO线程上执行

message：只有请求响应消息派发到线程池，其他的在IO线程上执行

execution：只有请求消息派发到线程池，响应消息和其他消息在IO线程上执行

connection：连接断开事件在IO上执行，其他的派发到线程池

threadpool

fixed

cached

limited

eager：优先创建worker线程

**直连提供者**：三种配置方式，一般只在测试环境使用

**只订阅**

<dubbo:registry address="10.20.153.10:9090" register="false" />

<dubbo:registry address="10.20.153.10:9090?register=false" />

**只注册**

<dubbo:registry id="qdRegistry" address="10.20.141.150:9090" subscribe="false" />

<dubbo:registry id="qdRegistry" address="10.20.141.150:9090?subscribe=false" />

**多协议**：不同服务不同协议。大数据用短连接协议，小数据大并发用长连接协议

不同服务配置不同协议

<!-- 多协议配置 -->

<dubbo:protocol name="dubbo" port="20880" />

<dubbo:protocol name="rmi" port="1099" />

<!-- 使用dubbo协议暴露服务 -->

<dubbo:service interface="com.alibaba.hello.api.HelloService" version="1.0.0" ref="helloService" protocol="dubbo" />

<!-- 使用rmi协议暴露服务 -->

<dubbo:service interface="com.alibaba.hello.api.DemoService" version="1.0.0" ref="demoService" protocol="rmi" />

同一服务配置多种协议

<!-- 多协议配置 -->

<dubbo:protocol name="dubbo" port="20880" />

<dubbo:protocol name="hessian" port="8080" />

<!-- 使用多个协议暴露服务 -->

<dubbo:service id="helloService" interface="com.alibaba.hello.api.HelloService" version="1.0.0" protocol="dubbo,hessian" />

**多注册中心**

一个服务注册到多个注册中心

<!-- 多注册中心配置 -->

<dubbo:registry id="hangzhouRegistry" address="10.20.141.150:9090" />

<dubbo:registry id="qingdaoRegistry" address="10.20.141.151:9010" default="false" />

<!-- 向多个注册中心注册 -->

<dubbo:service interface="com.alibaba.hello.api.HelloService" version="1.0.0" ref="helloService" registry="hangzhouRegistry,qingdaoRegistry" />

不同服务使用不同的注册中心

<!-- 多注册中心配置 -->

<dubbo:registry id="chinaRegistry" address="10.20.141.150:9090" />

<dubbo:registry id="intlRegistry" address="10.20.154.177:9010" default="false" />

<!-- 向中文站注册中心注册 -->

<dubbo:service interface="com.alibaba.hello.api.HelloService" version="1.0.0" ref="helloService" registry="chinaRegistry" />

<!-- 向国际站注册中心注册 -->

<dubbo:service interface="com.alibaba.hello.api.DemoService" version="1.0.0" ref="demoService" registry="intlRegistry" />

引用多注册中心

<!-- 多注册中心配置 -->

<dubbo:registry id="chinaRegistry" address="10.20.141.150:9090" />

<dubbo:registry id="intlRegistry" address="10.20.154.177:9010" default="false" />

<!-- 引用中文站服务 -->

<dubbo:reference id="chinaHelloService" interface="com.alibaba.hello.api.HelloService" version="1.0.0" registry="chinaRegistry" />

<!-- 引用国际站站服务 -->

<dubbo:reference id="intlHelloService" interface="com.alibaba.hello.api.HelloService" version="1.0.0" registry="intlRegistry" />

使用多注册中心

<!-- 多注册中心配置，竖号分隔表示同时连接多个不同注册中心，同一注册中心的多个集群地址用逗号分隔 -->

<dubbo:registry address="10.20.141.150:9090|10.20.154.177:9010" />

<!-- 引用服务 -->

<dubbo:reference id="helloService" interface="com.alibaba.hello.api.HelloService" version="1.0.0" />

**服务分组**：一个接口有多种实现

**多版本**：服务不兼容升级。

版本迁移步骤：

在低压力时间段，先升级一半提供者为新版本

再将所有消费者升级为新版本

然后将剩下的一半提供者升级为新版本

**分组聚合**：按组合并返回结果

搜索所有分组

<dubbo:reference interface="com.xxx.MenuService" group="\*" merger="true" />

合并指定分组

<dubbo:reference interface="com.xxx.MenuService" group="aaa,bbb" merger="true" />

指定方法合并结果，其它未指定的方法，将只调用一个 Group

<dubbo:reference interface="com.xxx.MenuService" group="\*">

<dubbo:method name="getMenuItems" merger="true" />

</dubbo:reference>

某个方法不合并结果，其它都合并结果

<dubbo:reference interface="com.xxx.MenuService" group="\*" merger="true">

<dubbo:method name="getMenuItems" merger="false" />

</dubbo:reference>

指定合并策略，缺省根据返回值类型自动匹配，如果同一类型有两个合并器时，需指定合并器的名称

<dubbo:reference interface="com.xxx.MenuService" group="\*">

<dubbo:method name="getMenuItems" merger="mymerge" />

</dubbo:reference>

指定合并方法，将调用返回结果的指定方法进行合并，合并方法的参数类型必须是返回结果类型本身

<dubbo:reference interface="com.xxx.MenuService" group="\*">

<dubbo:method name="getMenuItems" merger=".addAll" />

</dubbo:reference>

**参数验证**

**结果缓存**：缓存类型：lru、threadlocal、jcache

<dubbo:reference interface="com.foo.BarService" cache="lru" />

或：

<dubbo:reference interface="com.foo.BarService">

<dubbo:method name="findBar" cache="lru" />

</dubbo:reference>

**上下文信息**RpcContext

**隐式参数**：可以通过 RpcContext 上的 setAttachment 和 getAttachment 在服务消费方和提供方之间进行参数的隐式传递

**consumer异步调用**

CompletableFuture

**provider异步执行**

**参数回调**

**本地存根**

**本地伪装**：在出现 RpcException 时进行容错

<dubbo:reference interface="com.foo.BarService" mock="com.foo.BarServiceMock" />

<dubbo:reference interface="com.foo.BarService" mock="true" />

**延迟暴露**

**并发控制**

<dubbo:service interface="com.foo.BarService" executes="10" />

<dubbo:service interface="com.foo.BarService">

<dubbo:method name="sayHello" executes="10" />

</dubbo:service>

<dubbo:service interface="com.foo.BarService" actives="10" />

<dubbo:reference interface="com.foo.BarService" actives="10" />

**连接控制**

<dubbo:provider protocol="dubbo" accepts="10" />

<dubbo:protocol name="dubbo" accepts="10" />

<dubbo:reference interface="com.foo.BarService" connections="10" />

<dubbo:service interface="com.foo.BarService" connections="10" />