**使用指南**

注：详细文档可查看<http://code.alibabatech.com/wiki/display/dubbo/Home-zh> 以下内容只是对接的一个主要内容的摘要

* [安装手册](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh#AdministratorGuide-zh-%E5%AE%89%E8%A3%85%E6%89%8B%E5%86%8C)
  + [示例提供者安装](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh#AdministratorGuide-zh-%E7%A4%BA%E4%BE%8B%E6%8F%90%E4%BE%9B%E8%80%85%E5%AE%89%E8%A3%85)
  + [示例消费者安装](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh#AdministratorGuide-zh-%E7%A4%BA%E4%BE%8B%E6%B6%88%E8%B4%B9%E8%80%85%E5%AE%89%E8%A3%85)
  + [Zookeeper注册中心安装](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh#AdministratorGuide-zh-Zookeeper%E6%B3%A8%E5%86%8C%E4%B8%AD%E5%BF%83%E5%AE%89%E8%A3%85)
  + [Redis注册中心安装](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh#AdministratorGuide-zh-Redis%E6%B3%A8%E5%86%8C%E4%B8%AD%E5%BF%83%E5%AE%89%E8%A3%85)
  + [简易注册中心安装](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh#AdministratorGuide-zh-%E7%AE%80%E6%98%93%E6%B3%A8%E5%86%8C%E4%B8%AD%E5%BF%83%E5%AE%89%E8%A3%85)
  + [简易监控中心安装](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh#AdministratorGuide-zh-%E7%AE%80%E6%98%93%E7%9B%91%E6%8E%A7%E4%B8%AD%E5%BF%83%E5%AE%89%E8%A3%85)
  + [管理控制台安装](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh#AdministratorGuide-zh-%E7%AE%A1%E7%90%86%E6%8E%A7%E5%88%B6%E5%8F%B0%E5%AE%89%E8%A3%85)
* [运维手册](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh#AdministratorGuide-zh-%E8%BF%90%E7%BB%B4%E6%89%8B%E5%86%8C)
  + [管理控制台运维](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh#AdministratorGuide-zh-%E7%AE%A1%E7%90%86%E6%8E%A7%E5%88%B6%E5%8F%B0%E8%BF%90%E7%BB%B4)

|  |  |
| --- | --- |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | 推荐使用Zookeeper注册中心 |

* 你可以只运行[Demo Provider](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh#AdministratorGuide-zh-DemoProviderInstallation)和[Demo Consumer](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh#AdministratorGuide-zh-DemoConsumerInstallation)，它们缺省配置为通过Multicast注册中心广播互相发现，建议在不同机器上运行，如果在同一机器上，需设置unicast=false：即：multicast://224.5.6.7:1234?unicast=false，否则发给消费者的单播消息可能被提供者抢占，两个消费者在同一台机器也一样，只有multicast注册中心有此问题。
* 你也可以运行多个[Demo Provider](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh#AdministratorGuide-zh-%E7%A4%BA%E4%BE%8B%E6%8F%90%E4%BE%9B%E8%80%85%E5%AE%89%E8%A3%85)和[Demo Consumer](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh#AdministratorGuide-zh-%E7%A4%BA%E4%BE%8B%E6%B6%88%E8%B4%B9%E8%80%85%E5%AE%89%E8%A3%85)，来验证软负载均衡，Demo Consumer可以直接启动多个实例，而多个Demo Provider因有端口冲突，可在不同机器上运行，或者修改Demo Provider安装目录下conf/dubbo.properties配置中的dubbo.protocol.port的值。
* 你也可以增加运行[Simple Monitor](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh#AdministratorGuide-zh-%E7%AE%80%E6%98%93%E7%9B%91%E6%8E%A7%E4%B8%AD%E5%BF%83%E5%AE%89%E8%A3%85)监控中心，它缺省配置为通过Multicast注册中心广播发现Provider和Consumer，并展示出它们的依赖关系，以及它们之间调用的次数和时间。
* 你也可以将Multicast注册中心换成Zookeeper注册中心，安装[Zookeeper Registry](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh#AdministratorGuide-zh-Zookeeper%E6%B3%A8%E5%86%8C%E4%B8%AD%E5%BF%83%E5%AE%89%E8%A3%85)后，修改Demo Proivder，Demo Consumer，Simple Monitor三者安装目录下的conf/dubbo.properties，将dubbo.registry.address的值改为zookeeper://127.0.0.1:2181，同理，如果换成[Redis Registry](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh#AdministratorGuide-zh-Redis%E6%B3%A8%E5%86%8C%E4%B8%AD%E5%BF%83%E5%AE%89%E8%A3%85)，值改为redis://127.0.0.1:6379，如果换成[Simple Registry](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh#AdministratorGuide-zh-%E7%AE%80%E6%98%93%E6%B3%A8%E5%86%8C%E4%B8%AD%E5%BF%83%E5%AE%89%E8%A3%85)，值改为dubbo://127.0.0.1:9090  
  注意：multicast地址不能配成127.0.0.1，也不能配成机器的IP地址，必须是D段广播地址，也就是：224.0.0.0到239.255.255.255之间的任意地址

**安装手册**

([+](http://code.alibabatech.com/wiki/display/dubbo/Installation+Tutorial-zh)) ([#](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh#AdministratorGuide-zh-%E5%AE%89%E8%A3%85%E6%89%8B%E5%86%8C))

**示例提供者安装**

([+](http://code.alibabatech.com/wiki/display/dubbo/Demo+Provider+Installation-zh)) ([#](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh#AdministratorGuide-zh-%E7%A4%BA%E4%BE%8B%E6%8F%90%E4%BE%9B%E8%80%85%E5%AE%89%E8%A3%85))

**安装:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| wget http://code.alibabatech.com/mvn/releases/com/alibaba/dubbo-demo-provider/2.4.1/dubbo-demo-provider-2.4.1-assembly.tar.gz  tar zxvf dubbo-demo-provider-2.4.1-assembly.tar.gz  cd dubbo-demo-provider-2.4.1 |

**配置:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| vi conf/dubbo.properties |

**启动:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| ./bin/start.sh |

**停止:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| ./bin/stop.sh |

**重启:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| ./bin/restart.sh |

**调试:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| ./bin/start.sh debug |

**系统状态:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| ./bin/dump.sh |

**总控入口:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| ./bin/server.sh start  ./bin/server.sh stop  ./bin/server.sh restart  ./bin/server.sh debug  ./bin/server.sh dump |

**标准输出:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| tail -f logs/stdout.log |

**命令行:** (See: [Telnet Command Reference](http://code.alibabatech.com/wiki/display/dubbo/User+Guide#UserGuide-TelnetCommandReference))

telnet 127.0.0.1 20880  
help

Or:

echo status | nc -i 1 127.0.0.1 20880

**示例消费者安装**

([+](http://code.alibabatech.com/wiki/display/dubbo/Demo+Consumer+Installation-zh)) ([#](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh#AdministratorGuide-zh-%E7%A4%BA%E4%BE%8B%E6%B6%88%E8%B4%B9%E8%80%85%E5%AE%89%E8%A3%85))

**安装:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| wget http://code.alibabatech.com/mvn/releases/com/alibaba/dubbo-demo-consumer/2.4.1/dubbo-demo-consumer-2.4.1-assembly.tar.gz  tar zxvf dubbo-demo-consumer-2.4.1-assembly.tar.gz  cd dubbo-demo-consumer-2.4.1 |

**配置:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| vi conf/dubbo.properties |

**启动:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| ./bin/start.sh  tail -f logs/stdout.log |

**停止:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| ./bin/stop.sh |

**重启:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| ./bin/restart.sh |

**调试:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| ./bin/start.sh debug |

**系统状态:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| ./bin/dump.sh |

**总控入口:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| ./bin/server.sh start  ./bin/server.sh stop  ./bin/server.sh restart  ./bin/server.sh debug  ./bin/server.sh dump |

**标准输出:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| tail -f logs/stdout.log |

**Zookeeper注册中心安装**

([+](http://code.alibabatech.com/wiki/display/dubbo/Zookeeper+Registry+Installation-zh)) ([#](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh#AdministratorGuide-zh-Zookeeper%E6%B3%A8%E5%86%8C%E4%B8%AD%E5%BF%83%E5%AE%89%E8%A3%85))

|  |  |  |
| --- | --- | --- |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | | 建议使用dubbo-2.3.3以上版本的zookeeper注册中心客户端 |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | [Zookeeper](http://zookeeper.apache.org)是Apache Hadoop的子项目，强度相对较好，建议生产环境使用该注册中心。 | |

|  |  |
| --- | --- |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | Dubbo未对Zookeeper服务器端做任何侵入修改，只需安装原生的Zookeeper服务器即可，所有注册中心逻辑适配都在调用Zookeeper客户端时完成。 |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | 如果需要，可以考虑使用taobao的zookeeper监控：<http://rdc.taobao.com/team/jm/archives/1450>  但是此监控必须在linux环境下才能运行，里面的监控时用ssh的调用的，所以windows要用必须要有相应的配套ssh环境 |

**安装:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| wget http://www.apache.org/dist//zookeeper/zookeeper-3.3.3/zookeeper-3.3.3.tar.gz  tar zxvf zookeeper-3.3.3.tar.gz  cd zookeeper-3.3.3  cp conf/zoo\_sample.cfg conf/zoo.cfg |

**配置:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| vi conf/zoo.cfg |

如果不需要集群，zoo.cfg的内容如下：(其中data目录需改成你真实输出目录)

**zoo.cfg**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| tickTime=2000  initLimit=10  syncLimit=5  dataDir=/home/dubbo/zookeeper-3.3.3/data  clientPort=2181 |

如果需要集群，zoo.cfg的内容如下：(其中data目录和server地址需改成你真实部署机器的信息)

**zoo.cfg**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| tickTime=2000  initLimit=10  syncLimit=5  dataDir=/home/dubbo/zookeeper-3.3.3/data  clientPort=2181  server.1=10.20.153.10:2555:3555  server.2=10.20.153.11:2555:3555 |

并在data目录下放置myid文件：(上面zoo.cfg中的dataDir)

**注意：集群必须是2n+1 也就是说集群必须是三台才能成**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| mkdir data  vi myid |

myid指明自己的id，对应上面zoo.cfg中server.后的数字，第一台的内容为1，第二台的内容为2，内容如下：

**myid**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| 1 |

**启动:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| ./bin/zkServer.sh start |

**停止:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| ./bin/zkServer.sh stop |

**命令行:** (See: <http://zookeeper.apache.org/doc/r3.3.3/zookeeperAdmin.html>)

telnet 127.0.0.1 2181  
dump

Or:

echo dump | nc 127.0.0.1 2181

**用法:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| dubbo.registry.address=zookeeper://10.20.153.10:2181?backup=10.20.153.11:2181 |

Or:

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| <dubbo:registry protocol="zookeeper" address="10.20.153.10:2181,10.20.153.11:2181" /> |

**Redis注册中心安装**

([+](http://code.alibabatech.com/wiki/display/dubbo/Redis+Registry+Installation-zh)) ([#](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh#AdministratorGuide-zh-Redis%E6%B3%A8%E5%86%8C%E4%B8%AD%E5%BF%83%E5%AE%89%E8%A3%85))

|  |  |  |
| --- | --- | --- |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | | **Redis说明** Redis是一个高效的KV存储服务器，参见：<http://redis.io> |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | **Redis使用** 使用方式参见: [Redis使用手册](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-RedisRegistry)，只需搭一个原生的Redis服务器，并将[Quick Start](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-QuickStart)中Provider和Consumer里的conf/dubbo.properties中的dubbo.registry.addrss的值改为redis://127.0.0.1:6379即可使用 | |

|  |  |  |
| --- | --- | --- |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | **Redis集群** Redis注册中心集群采用在客户端同时写入多个服务器，读取单个服务器的策略实现。 | |
| http://code.alibabatech.com/wiki/images/icons/emoticons/warning.gif | | 2.1.0以上版本支持 |

参见：<http://redis.io/topics/quickstart>

**安装:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| wget http://redis.googlecode.com/files/redis-2.4.8.tar.gz  tar xzf redis-2.4.8.tar.gz  cd redis-2.4.8  make |

**配置:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| vi redis.conf |

**启动:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| nohup ./src/redis-server redis.conf & |

**停止:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| killall redis-server |

**命令行:** (参见: <http://redis.io/commands>)

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| ./src/redis-cli  hgetall /dubbo/com.foo.BarService/providers |

或者：

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| telnet 127.0.0.1 6379  hgetall /dubbo/com.foo.BarService/providers |

**简易注册中心安装**

([+](http://code.alibabatech.com/wiki/display/dubbo/Simple+Registry+Installation-zh)) ([#](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh#AdministratorGuide-zh-%E7%AE%80%E6%98%93%E6%B3%A8%E5%86%8C%E4%B8%AD%E5%BF%83%E5%AE%89%E8%A3%85))

|  |  |
| --- | --- |
| http://code.alibabatech.com/wiki/images/icons/emoticons/warning.gif | Simple Registry没有经过严格测试，可能不健状，并且不支持集群，不建议用于生产环境。 |

**安装:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| wget http://code.alibabatech.com/mvn/releases/com/alibaba/dubbo-registry-simple/2.4.1/dubbo-registry-simple-2.4.1-assembly.tar.gz  tar zxvf dubbo-registry-simple-2.4.1-assembly.tar.gz  cd dubbo-registry-simple-2.4.1 |

**配置:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| vi conf/dubbo.properties |

**启动:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| ./bin/start.sh |

**停止:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| ./bin/stop.sh |

**重启:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| ./bin/restart.sh |

**调试:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| ./bin/start.sh debug |

**系统状态:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| ./bin/dump.sh |

**总控入口:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| ./bin/server.sh start  ./bin/server.sh stop  ./bin/server.sh restart  ./bin/server.sh debug  ./bin/server.sh dump |

**标准输出:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| tail -f logs/stdout.log |

**命令行:** (See: [Telnet Command Reference](http://code.alibabatech.com/wiki/display/dubbo/User+Guide#UserGuide-TelnetCommandReference))

telnet 127.0.0.1 9090  
help

Or:

echo status | nc -i 1 127.0.0.1 9090

**简易监控中心安装**

([+](http://code.alibabatech.com/wiki/display/dubbo/Simple+Monitor+Installation-zh)) ([#](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh#AdministratorGuide-zh-%E7%AE%80%E6%98%93%E7%9B%91%E6%8E%A7%E4%B8%AD%E5%BF%83%E5%AE%89%E8%A3%85))

|  |  |
| --- | --- |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | Simple Monitor挂掉不会影响到Consumer和Provider之间的调用，所以用于生产环境不会有风险。 |
| http://code.alibabatech.com/wiki/images/icons/emoticons/warning.gif | Simple Monitor采用磁盘存储统计信息，请注意安装机器的磁盘限制，如果要集群，建议用[mount](http://linux.die.net/man/8/mount)共享磁盘。 |

|  |  |
| --- | --- |
| http://code.alibabatech.com/wiki/images/icons/emoticons/warning.gif | charts目录必须放在jetty.directory下，否则页面上访问不了。 |

**安装:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| wget http://code.alibabatech.com/mvn/releases/com/alibaba/dubbo-monitor-simple/2.4.1/dubbo-monitor-simple-2.4.1-assembly.tar.gz  tar zxvf dubbo-monitor-simple-2.4.1-assembly.tar.gz  cd dubbo-monitor-simple-2.4.1 |

**配置:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| vi conf/dubbo.properties |

**启动:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| ./bin/start.sh |

**停止:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| ./bin/stop.sh |

**重启:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| ./bin/restart.sh |

**调试:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| ./bin/start.sh debug |

**系统状态:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| ./bin/dump.sh |

**总控入口:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| ./bin/server.sh start  ./bin/server.sh stop  ./bin/server.sh restart  ./bin/server.sh debug  ./bin/server.sh dump |

**标准输出:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| tail -f logs/stdout.log |

**命令行:** (See: [Telnet Command Reference](http://code.alibabatech.com/wiki/display/dubbo/User+Guide#UserGuide-TelnetCommandReference))

telnet 127.0.0.1 7070  
help

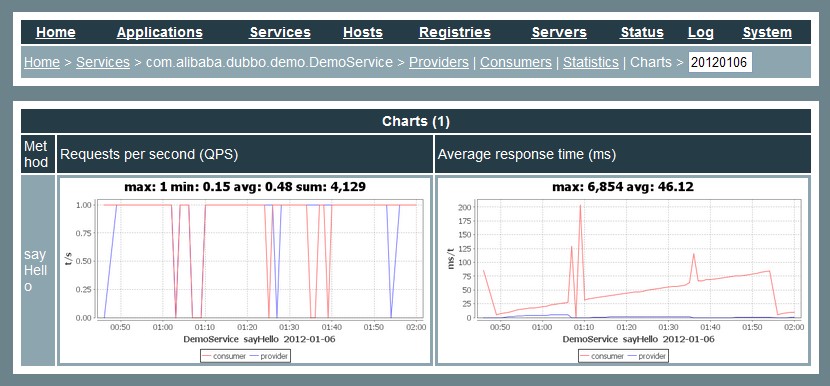
Or:

echo status | nc -i 1 127.0.0.1 7070

**访问:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| <http://127.0.0.1:8080> |



**管理控制台安装**

([+](http://code.alibabatech.com/wiki/display/dubbo/Admin+Console+Installation-zh)) ([#](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh#AdministratorGuide-zh-%E7%AE%A1%E7%90%86%E6%8E%A7%E5%88%B6%E5%8F%B0%E5%AE%89%E8%A3%85))

|  |  |
| --- | --- |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | 管理控制台为内部裁剪版本，开源部分主要包含：路由规则，动态配置，服务降级，访问控制，权重调整，负载均衡，等管理功能。 |

**安装:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| wget http://apache.etoak.com/tomcat/tomcat-6/v6.0.35/bin/apache-tomcat-6.0.35.tar.gz  tar zxvf apache-tomcat-6.0.35.tar.gz  cd apache-tomcat-6.0.35  rm -rf webapps/ROOT |

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| wget http://code.alibabatech.com/mvn/releases/com/alibaba/dubbo-admin/2.4.1/dubbo-admin-2.4.1.war  unzip dubbo-admin-2.4.1.war -d webapps/ROOT |

**配置:** (或将dubbo.properties放在当前用户目录下)

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| vi webapps/ROOT/WEB-INF/dubbo.properties |

**dubbo.properties**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| dubbo.registry.address=zookeeper://127.0.0.1:2181  dubbo.admin.root.password=root  dubbo.admin.guest.password=guest |

**启动:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| ./bin/startup.sh |

**停止:**

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| ./bin/shutdown.sh |

**访问:** (用户:root,密码:root 或 用户:guest,密码:guest)

[?](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh)

|  |
| --- |
| <http://127.0.0.1:8080/> |

运维手册

([+](http://code.alibabatech.com/wiki/display/dubbo/Operation+Tutorial-zh)) ([#](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh#AdministratorGuide-zh-%E8%BF%90%E7%BB%B4%E6%89%8B%E5%86%8C))

**管理控制台运维**

([+](http://code.alibabatech.com/wiki/display/dubbo/Admin+Console+Operation-zh)) ([#](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh#AdministratorGuide-zh-%E7%AE%A1%E7%90%86%E6%8E%A7%E5%88%B6%E5%8F%B0%E8%BF%90%E7%BB%B4))

**搜索页面**

当你需要管理Dubbo的服务时，首先要搜索到这个服务，然后打开它的管理页面：



**服务提供者页面**



**服务消费者页面**



**服务应用页面**



**添加路由规则页面**



**添加动态配置页面**



**Dubbo用户比较**

**DUBBO用法与比较**

([#](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-%E7%94%A8%E6%B3%95))

**本地服务：(Spring配置)**

**local.xml**

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <bean id=“xxxService” class=“com.xxx.XxxServiceImpl” />  <bean id=“xxxAction” class=“com.xxx.XxxAction”>  <property name=“xxxService” ref=“xxxService” />  </bean> |

**Dubbo 远程服务：(Spring配置)**

在本地服务的基础上，只需做简单配置，即可完成远程化：

* 将上面的local.xml配置拆分成两份，将服务定义部分放在服务提供方remote-provider.xml，将服务引用部分放在服务消费方remote-consumer.xml。
* 并在提供方增加暴露服务配置<dubbo:service>，在消费方增加引用服务配置<dubbo:reference>。

如下：

**remote-provider.xml**

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <bean id=“xxxService” class=“com.xxx.XxxServiceImpl” /> <!-- 和本地服务一样实现远程服务 -->  <dubbo:service interface=“com.xxx.XxxService” ref=“xxxService” /> <!-- 增加暴露远程服务配置 --> |

**remote-consumer.xml**

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:reference id=“xxxService” interface=“com.xxx.XxxService” /> <!-- 增加引用远程服务配置 -->  <bean id=“xxxAction” class=“com.xxx.XxxAction”> <!-- 和本地服务一样使用远程服务 -->  <property name=“xxxService” ref=“xxxService” />  </bean> |

**Hession 远程服务：(Spring配置)**

在本地服务的基础上，只需做简单配置，即可完成远程化：

* 将上面的local.xml配置拆分成两份，将服务定义部分放在服务提供方remote-provider.xml，将服务引用部分放在服务消费方remote-consumer.xml。
* 并在提供方增加暴露服务配置<dubbo:service>，在消费方增加引用服务配置<dubbo:reference>。

如下：

**remote-provider.xml**

**目前orderService 配置 remoting-servlet.xml**

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <bean id=“xxxService” class=“com.xxx.XxxServiceImpl” /> <!-- 和本地服务一样实现远程服务 -->  <bean name="/xxxService"class="org.springframework.remoting.caucho.HessianServiceExporter">  <property name="service" ref="xxxService" />  <property name="serviceInterface" value="com.xxx.XxxService " />  </bean> |

**remote-consumer.xml**

**目前调用orderService 配置 applicationContext.xml**

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <bean  name="/xxxService" class="org.springframework.remoting.caucho.HessianProxyFactoryBean">  <property name="serviceUrl" value="${order\_server}/xxxService"/>  <property name="serviceInterface" value="com.xxx.XxxService"/>  </bean> |

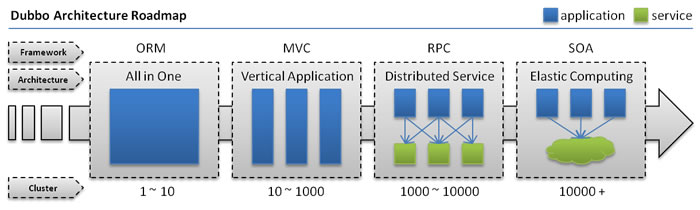
**Dubbo入门**

([+](http://code.alibabatech.com/wiki/display/dubbo/Getting+Started-zh)) ([#](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-%E5%85%A5%E9%97%A8))

**背景**

([#](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-%E8%83%8C%E6%99%AF))

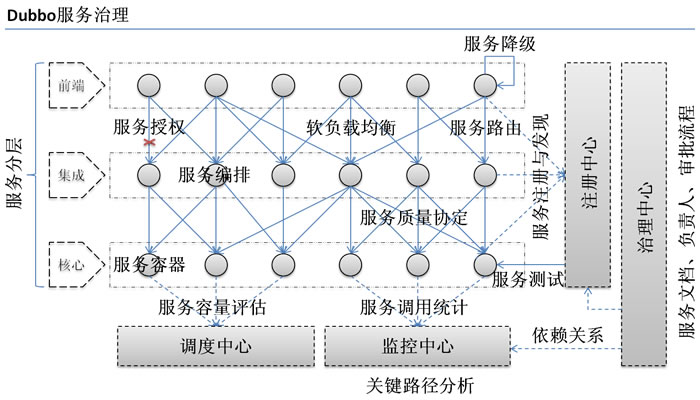
随着互联网的发展，网站应用的规模不断扩大，常规的垂直应用架构已无法应对，分布式服务架构以及流动计算架构势在必行，亟需一个治理系统确保架构有条不紊的演进。



* **单一应用架构**
  + 当网站流量很小时，只需一个应用，将所有功能都部署在一起，以减少部署节点和成本。
  + 此时，用于简化增删改查工作量的 **数据访问框架(ORM)** 是关键。
* **垂直应用架构**
  + 当访问量逐渐增大，单一应用增加机器带来的加速度越来越小，将应用拆成互不相干的几个应用，以提升效率。
  + 此时，用于加速前端页面开发的 **Web框架(MVC)** 是关键。
* **分布式服务架构**
  + 当垂直应用越来越多，应用之间交互不可避免，将核心业务抽取出来，作为独立的服务，逐渐形成稳定的服务中心，使前端应用能更快速的响应多变的市场需求。
  + 此时，用于提高业务复用及整合的 **分布式服务框架(RPC)** 是关键。
* **流动计算架构**
  + 当服务越来越多，容量的评估，小服务资源的浪费等问题逐渐显现，此时需增加一个调度中心基于访问压力实时管理集群容量，提高集群利用率。
  + 此时，用于提高机器利用率的 **资源调度和治理中心(SOA)** 是关键。

**需求**

([#](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-%E9%9C%80%E6%B1%82))



在大规模服务化之前，应用可能只是通过RMI或Hessian等工具，简单的暴露和引用远程服务，通过配置服务的URL地址进行调用，通过F5等硬件进行负载均衡。

**(1) 当服务越来越多时，服务URL配置管理变得非常困难，F5硬件负载均衡器的单点压力也越来越大。**

此时需要一个服务注册中心，动态的注册和发现服务，使服务的位置透明。

并通过在消费方获取服务提供方地址列表，实现软负载均衡和Failover，降低对F5硬件负载均衡器的依赖，也能减少部分成本。

**(2) 当进一步发展，服务间依赖关系变得错踪复杂，甚至分不清哪个应用要在哪个应用之前启动，架构师都不能完整的描述应用的架构关系。**

这时，需要自动画出应用间的依赖关系图，以帮助架构师理清理关系。

**(3) 接着，服务的调用量越来越大，服务的容量问题就暴露出来，这个服务需要多少机器支撑？什么时候该加机器？**

为了解决这些问题，第一步，要将服务现在每天的调用量，响应时间，都统计出来，作为容量规划的参考指标。

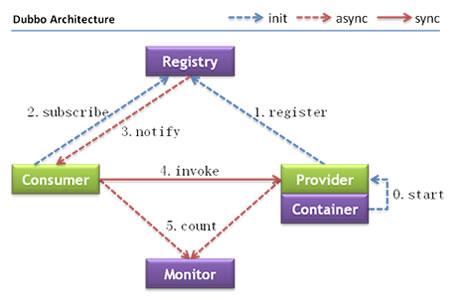
其次，要可以动态调整权重，在线上，将某台机器的权重一直加大，并在加大的过程中记录响应时间的变化，直到响应时间到达阀值，记录此时的访问量，再以此访问量乘以机器数反推总容量。

以上是Dubbo最基本的几个需求，更多服务治理问题参见：

<http://code.alibabatech.com/blog/experience_1402/service-governance-process.html>

**架构**

([#](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-%E6%9E%B6%E6%9E%84))



**节点角色说明：**

* **Provider:** 暴露服务的服务提供方。
* **Consumer:** 调用远程服务的服务消费方。
* **Registry:** 服务注册与发现的注册中心。
* **Monitor:** 统计服务的调用次调和调用时间的监控中心。
* **Container:** 服务运行容器。

**调用关系说明：**

* 0. 服务容器负责启动，加载，运行服务提供者。
* 1. 服务提供者在启动时，向注册中心注册自己提供的服务。
* 2. 服务消费者在启动时，向注册中心订阅自己所需的服务。
* 3. 注册中心返回服务提供者地址列表给消费者，如果有变更，注册中心将基于长连接推送变更数据给消费者。
* 4. 服务消费者，从提供者地址列表中，基于软负载均衡算法，选一台提供者进行调用，如果调用失败，再选另一台调用。
* 5. 服务消费者和提供者，在内存中累计调用次数和调用时间，定时每分钟发送一次统计数据到监控中心。

**(1) 连通性：**

* 注册中心负责服务地址的注册与查找，相当于目录服务，服务提供者和消费者只在启动时与注册中心交互，注册中心不转发请求，压力较小
* 监控中心负责统计各服务调用次数，调用时间等，统计先在内存汇总后每分钟一次发送到监控中心服务器，并以报表展示
* 服务提供者向注册中心注册其提供的服务，并汇报调用时间到监控中心，此时间不包含网络开销
* 服务消费者向注册中心获取服务提供者地址列表，并根据负载算法直接调用提供者，同时汇报调用时间到监控中心，此时间包含网络开销
* 注册中心，服务提供者，服务消费者三者之间均为长连接，监控中心除外
* 注册中心通过长连接感知服务提供者的存在，服务提供者宕机，注册中心将立即推送事件通知消费者
* 注册中心和监控中心全部宕机，不影响已运行的提供者和消费者，消费者在本地缓存了提供者列表
* 注册中心和监控中心都是可选的，服务消费者可以直连服务提供者

**(2) 健状性：**

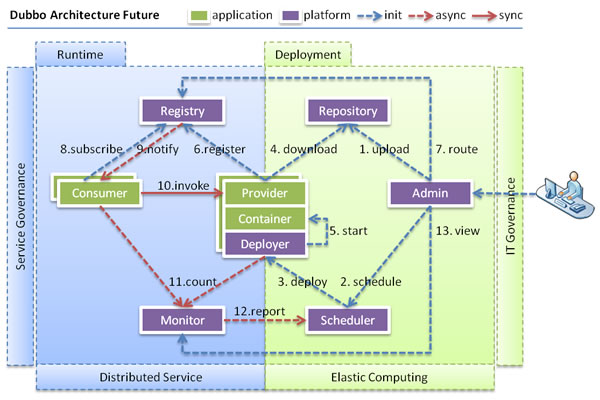
* 监控中心宕掉不影响使用，只是丢失部分采样数据
* 数据库宕掉后，注册中心仍能通过缓存提供服务列表查询，但不能注册新服务
* 注册中心对等集群，任意一台宕掉后，将自动切换到另一台
* 注册中心全部宕掉后，服务提供者和服务消费者仍能通过本地缓存通讯
* 服务提供者无状态，任意一台宕掉后，不影响使用
* 服务提供者全部宕掉后，服务消费者应用将无法使用，并无限次重连等待服务提供者恢复

**(3) 伸缩性：**

* 注册中心为对等集群，可动态增加机器部署实例，所有客户端将自动发现新的注册中心
* 服务提供者无状态，可动态增加机器部署实例，注册中心将推送新的服务提供者信息给消费者

**(4) 升级性：**

* 当服务集群规模进一步扩大，带动IT治理结构进一步升级，需要实现动态部署，进行流动计算，现有分布式服务架构不会带来阻力：



**依赖**

([+](http://code.alibabatech.com/wiki/display/dubbo/Dependencies-zh)) ([#](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-%E4%BE%9D%E8%B5%96))

**必需依赖**

* JDK1.5+

|  |  |
| --- | --- |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | 理论上Dubbo可以只依赖JDK，不依赖于任何三方库运行，只需配置使用JDK相关实现策略。 |

**缺省依赖**

通过mvn dependency:tree > dep.log命令分析，Dubbo缺省依赖以下三方库：

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| [INFO] +- com.alibaba:dubbo:jar:2.1.2:compile  [INFO] | +- log4j:log4j:jar:1.2.16:compile  [INFO] | +- org.javassist:javassist:jar:3.15.0-GA:compile  [INFO] | +- org.springframework:spring:jar:2.5.6.SEC03:compile  [INFO] | +- commons-logging:commons-logging:jar:1.1.1:compile  [INFO] | \- org.jboss.netty:netty:jar:3.2.5.Final:compile |

这里所有依赖都是换照Dubbo缺省配置选的，这些缺省值是基于稳定性和性能考虑的。

* log4j.jar和commons-logging.jar日志输出包。
  + 可以直接去掉，dubbo本身的日志会自动切换为JDK的java.util.logging输出。
  + 但如果其它三方库比如spring.jar间接依赖commons-logging，则不能去掉。
* javassist.jar 字节码生成。
  + 如果<dubbo:provider proxy="jdk" />或<dubbo:consumer proxy="jdk" />，以及<dubbo:application compiler="jdk" />，则不需要。
* spring.jar 配置解析。
  + 如果用ServiceConfig和ReferenceConfig的API调用，则不需要。
* netty.jar 网络传输。
  + 如果<dubbo:protocol server="mina"/>或<dubbo:protocol server="grizzly"/>，则换成mina.jar或grizzly.jar。
  + 如果<protocol name="rmi"/>，则不需要。

**可选依赖**

以下依赖，在主动配置使用相应实现策略时用到，需自行加入依赖。

* mina: 1.1.7
* grizzly: 2.1.4
* httpclient: 4.1.2
* hessian\_lite: 3.2.1-fixed
* xstream: 1.4.1
* fastjson: 1.1.8
* zookeeper: 3.3.3
* jedis: 2.0.0
* xmemcached: 1.3.6
* jfreechart: 1.0.13
* hessian: 4.0.7
* jetty: 6.1.26
* hibernate-validator: 4.2.0.Final
* zkclient: 0.1
* curator: 1.1.10
* cxf: 2.6.1
* thrift: 0.8.0

JEE:

* servlet: 2.5
* bsf: 3.1
* validation-api: 1.0.0.GA
* jcache: 0.4

**配置**

([+](http://code.alibabatech.com/wiki/display/dubbo/Configs-zh)) ([#](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-%E9%85%8D%E7%BD%AE))

**Xml配置**

([+](http://code.alibabatech.com/wiki/display/dubbo/Xml+Config-zh)) ([#](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-Xml%E9%85%8D%E7%BD%AE))

|  |  |  |
| --- | --- | --- |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | | **配置项说明** 详细配置项，请参见：[配置参考手册](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-%E9%85%8D%E7%BD%AE%E5%8F%82%E8%80%83%E6%89%8B%E5%86%8C) ([+](http://code.alibabatech.com/wiki/display/dubbo/Configuration+Reference-zh)) |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | **API使用说明** 如果不想使用Spring配置，而希望通过API的方式进行调用，请参见：[API配置](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-API%E9%85%8D%E7%BD%AE) ([+](http://code.alibabatech.com/wiki/display/dubbo/API+Config-zh)) | |

|  |  |
| --- | --- |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | **配置使用说明** 想知道如何使用配置，请参见：[快速启动](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-%E5%BF%AB%E9%80%9F%E5%90%AF%E5%8A%A8) ([+](http://code.alibabatech.com/wiki/display/dubbo/Quick+Start-zh)) |

示例：

**provider.xml**

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |  |
| --- | --- |
| <?xml version="1.0" encoding="UTF-8"?>  <beans xmlns="[http://www.springframework.org/schema/beans"](http://www.springframework.org/schema/beans)  xmlns:xsi="[http://www.w3.org/2001/XMLSchema-instance"](http://www.w3.org/2001/XMLSchema-instance)  xmlns:dubbo="[http://code.alibabatech.com/schema/dubbo"](http://code.alibabatech.com/schema/dubbo)  xsi:schemaLocation="http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans.xsd  http://code.alibabatech.com/schema/dubbo  http://code.alibabatech.com/schema/dubbo/dubbo.xsd  ">  <dubbo:application name="hello-world-app" />  <dubbo:registry address="multicast://224.5.6.7:1234" />  <dubbo:protocol name="dubbo" port="20880" />  <dubbo:service interface="com.alibaba.dubbo.demo.DemoService" ref="demoServiceLocal" />  <dubbo:reference id="demoServiceRemote" interface="com.alibaba.dubbo.demo.DemoService" />  </beans> | |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | 所有标签者支持自定义参数，用于不同扩展点实现的特殊配置。 |

如：

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:protocol name="jms">  <dubbo:parameter key="queue" value="10.20.31.22" />  </dubbo:protocol> |

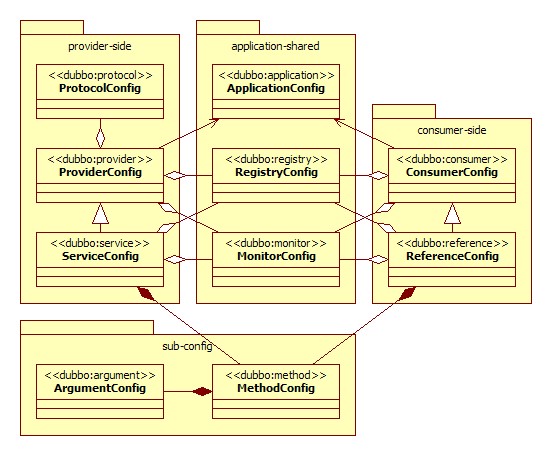
或：(2.1.0开始支持)

|  |  |
| --- | --- |
| http://code.alibabatech.com/wiki/images/icons/emoticons/warning.gif | 注意声明：xmlns:p="http://www.springframework.org/schema/p" |

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <beans xmlns="[http://www.springframework.org/schema/beans"](http://www.springframework.org/schema/beans)  xmlns:xsi="[http://www.w3.org/2001/XMLSchema-instance"](http://www.w3.org/2001/XMLSchema-instance)  xmlns:dubbo="[http://code.alibabatech.com/schema/dubbo"](http://code.alibabatech.com/schema/dubbo)  xmlns:p="[http://www.springframework.org/schema/p"](http://www.springframework.org/schema/p)  xsi:schemaLocation="http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans.xsd  http://code.alibabatech.com/schema/dubbo  http://code.alibabatech.com/schema/dubbo/dubbo.xsd  ">  <dubbo:protocol name="jms" p:queue="10.20.31.22" />  </beans> |

**Configuration Relation:**



* [<dubbo:service/>](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-%253Cdubbo%253Aservice%252F%253E) 服务配置，用于暴露一个服务，定义服务的元信息，一个服务可以用多个协议暴露，一个服务也可以注册到多个注册中心。
* [<dubbo:reference/>](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-%253Cdubbo%253Areference%252F%253E) 引用配置，用于创建一个远程服务代理，一个引用可以指向多个注册中心。
* [<dubbo:protocol/>](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-%253Cdubbo%253Aprotocol%252F%253E) 协议配置，用于配置提供服务的协议信息，协议由提供方指定，消费方被动接受。
* [<dubbo:application/>](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-%253Cdubbo%253Aapplication%252F%253E) 应用配置，用于配置当前应用信息，不管该应用是提供者还是消费者。
* [<dubbo:module/>](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-%253Cdubbo%253Amodule%252F%253E) 模块配置，用于配置当前模块信息，可选。
* [<dubbo:registry/>](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-%253Cdubbo%253Aregistry%252F%253E) 注册中心配置，用于配置连接注册中心相关信息。
* [<dubbo:monitor/>](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-%253Cdubbo%253Amonitor%252F%253E) 监控中心配置，用于配置连接监控中心相关信息，可选。
* [<dubbo:provider/>](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-%253Cdubbo%253Aprovider%252F%253E) 提供方的缺省值，当ProtocolConfig和ServiceConfig某属性没有配置时，采用此缺省值，可选。
* [<dubbo:consumer/>](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-%253Cdubbo%253Aconsumer%252F%253E) 消费方缺省配置，当ReferenceConfig某属性没有配置时，采用此缺省值，可选。
* [<dubbo:method/>](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-%253Cdubbo%253Amethod%252F%253E) 方法配置，用于ServiceConfig和ReferenceConfig指定方法级的配置信息。
* [<dubbo:argument/>](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-%253Cdubbo%253Aargument%252F%253E) 用于指定方法参数配置。

**Configuration Override:**



* 上图中以timeout为例，显示了配置的查找顺序，其它retries, loadbalance, actives等类似。
  + 方法级优先，接口级次之，全局配置再次之。
  + 如果级别一样，则消费方优先，提供方次之。
* 其中，服务提供方配置，通过URL经由注册中心传递给消费方。
* 建议由服务提供方设置超时，因为一个方法需要执行多长时间，服务提供方更清楚，如果一个消费方同时引用多个服务，就不需要关心每个服务的超时设置。
* 理论上ReferenceConfig的非服务标识配置，在ConsumerConfig，ServiceConfig, ProviderConfig均可以缺省配置。

**属性配置**

([+](http://code.alibabatech.com/wiki/display/dubbo/Properties+Config-zh)) ([#](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-%E5%B1%9E%E6%80%A7%E9%85%8D%E7%BD%AE))

|  |  |
| --- | --- |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | 如果公共配置很简单，没有多注册中心，多协议等情况，或者想多个Spring容器想共享配置，可以使用dubbo.properties作为缺省配置。 |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | Dubbo将自动加载classpath根目录下的dubbo.properties，可以通过JVM启动参数：-Ddubbo.properties.file=xxx.properties 改变缺省配置位置。 |

|  |  |
| --- | --- |
| http://code.alibabatech.com/wiki/images/icons/emoticons/warning.gif | 如果classpath根目录下存在多个dubbo.properties，比如多个jar包中有dubbo.properties，Dubbo会任意加载，并打印Error日志，后续可能改为抛异常。 |

映射规则：

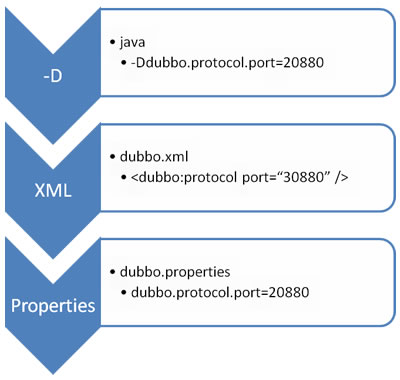
* 将XML配置的标签名，加属性名，用点分隔，多个属性拆成多行：
  + 比如：dubbo.application.name=foo等价于<dubbo:application name="foo" />
  + 比如：dubbo.registry.address=10.20.153.10:9090等价于<dubbo:registry address="10.20.153.10:9090" />
* 如果XML有多行同名标签配置，可用id号区分，如果没有id号将对所有同名标签生效：
  + 比如：dubbo.protocol.rmi.port=1234等价于<dubbo:protocol id="rmi" name="rmi" port="1099" /> (协议的id没配时，缺省使用协议名作为id)
  + 比如：dubbo.registry.china.address=10.20.153.10:9090等价于<dubbo:registry id="china" address="10.20.153.10:9090" />

典型配置如：

**dubbo.properties**

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| dubbo.application.name=foo  dubbo.application.owner=bar  dubbo.registry.address=10.20.153.10:9090 |



覆盖策略：

* JVM启动-D参数优先，这样可以使用户在部署和启动时进行参数重写，比如在启动时需改变协议的端口。
* XML次之，如果在XML中有配置，则dubbo.properties中的相应配置项无效。
* Properties最后，相当于缺省值，只有XML没有配置时，dubbo.properties的相应配置项才会生效，通常用于共享公共配置，比如应用名。

**注解配置**

([+](http://code.alibabatech.com/wiki/display/dubbo/Annotation+Config-zh)) ([#](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-%E6%B3%A8%E8%A7%A3%E9%85%8D%E7%BD%AE))

|  |  |
| --- | --- |
| http://code.alibabatech.com/wiki/images/icons/emoticons/warning.gif | 2.2.1以上版本支持 |

服务提供方注解：

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| import com.alibaba.dubbo.config.annotation.Service;  @Service(version="1.0.0")  public class FooServiceImpl implements FooService {  // ......  } |

服务提供方配置：

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <!-- 公共信息，也可以用dubbo.properties配置 -->  <dubbo:application name="annotation-provider" />  <dubbo:registry address="127.0.0.1:4548" />  <!-- 扫描注解包路径，多个包用逗号分隔，不填pacakge表示扫描当前ApplicationContext中所有的类 -->  <dubbo:annotation package="com.foo.bar.service" /> |

服务消费方注解：

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| import com.alibaba.dubbo.config.annotation.Reference;  import org.springframework.stereotype.Component;  @Component  public class BarAction {  @Reference(version="1.0.0")  private FooService fooService;  } |

服务消费方配置：

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <!-- 公共信息，也可以用dubbo.properties配置 -->  <dubbo:application name="annotation-consumer" />  <dubbo:registry address="127.0.0.1:4548" />  <!-- 扫描注解包路径，多个包用逗号分隔，不填pacakge表示扫描当前ApplicationContext中所有的类 -->  <dubbo:annotation package="com.foo.bar.action" /> |

也可以使用：(等价于前面的：<dubbo:annotation package="com.foo.bar.service" />)

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |  |
| --- | --- |
| <dubbo:annotation />  <context:component-scan base-package="com.foo.bar.service">  <context:include-filter type="annotation" expression="com.alibaba.dubbo.config.annotation.Service" />  </context:component-scan> | |
| http://code.alibabatech.com/wiki/images/icons/emoticons/warning.gif | Spring2.5及以后版本支持component-scan，如果用的是Spring2.0及以前版本，需配置：  [?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)   |  | | --- | | <!-- Spring2.0支持@Service注解配置，但不支持package属性自动加载bean的实例，需人工定义bean的实例。-->  <dubbo:annotation />  <bean id="barService" class="com.foo.BarServiceImpl" /> | |

**API配置**

([+](http://code.alibabatech.com/wiki/display/dubbo/API+Config-zh)) ([#](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-API%E9%85%8D%E7%BD%AE))

|  |  |  |
| --- | --- | --- |
| http://code.alibabatech.com/wiki/images/icons/emoticons/warning.gif | **API使用范围** API仅用于OpenAPI, ESB, Test, Mock等系统集成，普通服务提供方或消费方，请采用配置方式使用Dubbo，请参见：[Xml配置](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-Xml%E9%85%8D%E7%BD%AE) ([+](http://code.alibabatech.com/wiki/display/dubbo/Xml+Config-zh)) | |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | | **API属性含义参考** API属性与配置项一对一，各属性含义，请参见：[配置参考手册](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-%E9%85%8D%E7%BD%AE%E5%8F%82%E8%80%83%E6%89%8B%E5%86%8C) ([+](http://code.alibabatech.com/wiki/display/dubbo/Configuration+Reference-zh))， 比如：ApplicationConfig.setName("xxx") 对应 <dubbo:application name="xxx" /> |

**(1) 服务提供者：**

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| import com.alibaba.dubbo.rpc.config.ApplicationConfig;  import com.alibaba.dubbo.rpc.config.RegistryConfig;  import com.alibaba.dubbo.rpc.config.ProviderConfig;  import com.alibaba.dubbo.rpc.config.ServiceConfig;  import com.xxx.XxxService;  import com.xxx.XxxServiceImpl;  // 服务实现  XxxService xxxService = new XxxServiceImpl();  // 当前应用配置  ApplicationConfig application = new ApplicationConfig();  application.setName("xxx");  // 连接注册中心配置  RegistryConfig registry = new RegistryConfig();  registry.setAddress("10.20.130.230:9090");  registry.setUsername("aaa");  registry.setPassword("bbb");  // 服务提供者协议配置  ProtocolConfig protocol = new ProtocolConfig();  protocol.setName("dubbo");  protocol.setPort(12345);  protocol.setThreads(200);  // 注意：ServiceConfig为重对象，内部封装了与注册中心的连接，以及开启服务端口  // 服务提供者暴露服务配置  ServiceConfig<XxxService> service = new ServiceConfig<XxxService>(); // 此实例很重，封装了与注册中心的连接，请自行缓存，否则可能造成内存和连接泄漏  service.setApplication(application);  service.setRegistry(registry); // 多个注册中心可以用setRegistries()  service.setProtocol(protocol); // 多个协议可以用setProtocols()  service.setInterface(XxxService.class);  service.setRef(xxxService);  service.setVersion("1.0.0");  // 暴露及注册服务  service.export(); |

**(2) 服务消费者：**

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| import com.alibaba.dubbo.rpc.config.ApplicationConfig;  import com.alibaba.dubbo.rpc.config.RegistryConfig;  import com.alibaba.dubbo.rpc.config.ConsumerConfig;  import com.alibaba.dubbo.rpc.config.ReferenceConfig;  import com.xxx.XxxService;  // 当前应用配置  ApplicationConfig application = new ApplicationConfig();  application.setName("yyy");  // 连接注册中心配置  RegistryConfig registry = new RegistryConfig();  registry.setAddress("10.20.130.230:9090");  registry.setUsername("aaa");  registry.setPassword("bbb");  // 注意：ReferenceConfig为重对象，内部封装了与注册中心的连接，以及与服务提供方的连接  // 引用远程服务  ReferenceConfig<XxxService> reference = new ReferenceConfig<XxxService>(); // 此实例很重，封装了与注册中心的连接以及与提供者的连接，请自行缓存，否则可能造成内存和连接泄漏  reference.setApplication(application);  reference.setRegistry(registry); // 多个注册中心可以用setRegistries()  reference.setInterface(XxxService.class);  reference.setVersion("1.0.0");  // 和本地bean一样使用xxxService  XxxService xxxService = reference.get(); // 注意：此代理对象内部封装了所有通讯细节，对象较重，请缓存复用 |

**(3) 特殊场景**

注：下面只列出不同的地方，其它参见上面的写法

**(3.1) 方法级设置：**

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| ...  // 方法级配置  List<MethodConfig> methods = new ArrayList<MethodConfig>();  MethodConfig method = new MethodConfig();  method.setName("createXxx");  method.setTimeout(10000);  method.setRetries(0);  methods.add(method);  // 引用远程服务  ReferenceConfig<XxxService> reference = new ReferenceConfig<XxxService>(); // 此实例很重，封装了与注册中心的连接以及与提供者的连接，请自行缓存，否则可能造成内存和连接泄漏  ...  reference.setMethods(methods); // 设置方法级配置  ... |

**(3.2) 点对点直连：**

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| ...  ReferenceConfig<XxxService> reference = new ReferenceConfig<XxxService>(); // 此实例很重，封装了与注册中心的连接以及与提供者的连接，请自行缓存，否则可能造成内存和连接泄漏  // 如果点对点直连，可以用reference.setUrl()指定目标地址，设置url后将绕过注册中心，  // 其中，协议对应provider.setProtocol()的值，端口对应provider.setPort()的值，  // 路径对应service.setPath()的值，如果未设置path，缺省path为接口名  reference.setUrl("dubbo://10.20.130.230:20880/com.xxx.XxxService");  ... |

**示例**

([+](http://code.alibabatech.com/wiki/display/dubbo/Examples-zh)) ([#](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-%E7%A4%BA%E4%BE%8B))

|  |  |  |
| --- | --- | --- |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | | 想完整的运行起来，请参见：[快速启动](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-%E5%BF%AB%E9%80%9F%E5%90%AF%E5%8A%A8) ([+](http://code.alibabatech.com/wiki/display/dubbo/Quick+Start-zh))，这里只列出各种场景的配置方式 |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | 以下示例全部使用基于Spring的[Xml配置](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-Xml%E9%85%8D%E7%BD%AE) ([+](http://code.alibabatech.com/wiki/display/dubbo/Xml+Config-zh))作为参考，如果不想使用Spring，而希望通过API的方式进行调用，请参见：[API配置](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-API%E9%85%8D%E7%BD%AE) ([+](http://code.alibabatech.com/wiki/display/dubbo/API+Config-zh)) | |

**启动时检查**

([+](http://code.alibabatech.com/wiki/display/dubbo/Check+On+Startup-zh)) ([#](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-%E5%90%AF%E5%8A%A8%E6%97%B6%E6%A3%80%E6%9F%A5))

|  |  |
| --- | --- |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | Dubbo缺省会在启动时检查依赖的服务是否可用，不可用时会抛出异常，阻止Spring初始化完成，以便上线时，能及早发现问题，默认check=true。 |
| http://code.alibabatech.com/wiki/images/icons/emoticons/warning.gif | 如果你的Spring容器是懒加载的，或者通过API编程延迟引用服务，请关闭check，否则服务临时不可用时，会抛出异常，拿到null引用，如果check=false，总是会返回引用，当服务恢复时，能自动连上。 |

可以通过check="false"关闭检查，比如，测试时，有些服务不关心，或者出现了循环依赖，必须有一方先启动。

关闭某个服务的启动时检查：(没有提供者时报错)

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:reference interface="com.foo.BarService" check="false" /> |

关闭所有服务的启动时检查：(没有提供者时报错)

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:consumer check="false" /> |

关闭注册中心启动时检查：(注册订阅失败时报错)

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:registry check="false" /> |

也可以用dubbo.properties配置：

**dubbo.properties**

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| dubbo.reference.com.foo.BarService.check=false  dubbo.reference.check=false  dubbo.consumer.check=false  dubbo.registry.check=false |

也可以用-D参数：

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |  |
| --- | --- |
| java -Ddubbo.reference.com.foo.BarService.check=false  java -Ddubbo.reference.check=false  java -Ddubbo.consumer.check=false  java -Ddubbo.registry.check=false | |
| http://code.alibabatech.com/wiki/images/icons/emoticons/warning.gif | **注意区别**   * dubbo.reference.check=false，强制改变所有reference的check值，就算配置中有声明，也会被覆盖。 * dubbo.consumer.check=false，是设置check的缺省值，如果配置中有显式的声明，如：<dubbo:reference check="true"/>，不会受影响。 * dubbo.registry.check=false，前面两个都是指订阅成功，但提供者列表是否为空是否报错，如果注册订阅失败时，也允许启动，需使用此选项，将在后台定时重试。 |

引用缺省是延迟初始化的，只有引用被注入到其它Bean，或被getBean()获取，才会初始化。  
如果需要饥饿加载，即没有人引用也立即生成动态代理，可以配置：

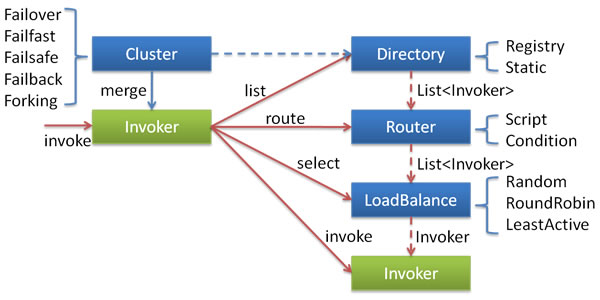
[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:reference interface="com.foo.BarService" init="true" /> |

**集群容错**

([+](http://code.alibabatech.com/wiki/display/dubbo/Fault+Tolerance-zh)) ([#](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-%E9%9B%86%E7%BE%A4%E5%AE%B9%E9%94%99))

|  |  |
| --- | --- |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | 在集群调用失败时，Dubbo提供了多种容错方案，缺省为failover重试。 |



各节点关系：

* 这里的Invoker是Provider的一个可调用Service的抽象，Invoker封装了Provider地址及Service接口信息。
* Directory代表多个Invoker，可以把它看成List<Invoker>，但与List不同的是，它的值可能是动态变化的，比如注册中心推送变更。
* Cluster将Directory中的多个Invoker伪装成一个Invoker，对上层透明，伪装过程包含了容错逻辑，调用失败后，重试另一个。
* Router负责从多个Invoker中按路由规则选出子集，比如读写分离，应用隔离等。
* LoadBalance负责从多个Invoker中选出具体的一个用于本次调用，选的过程包含了负载均衡算法，调用失败后，需要重选。

**集群容错模式：**

可以自行扩展集群容错策略，参见：[集群扩展](http://code.alibabatech.com/wiki/display/dubbo/Developer+Guide-zh#DeveloperGuide-zh-%E9%9B%86%E7%BE%A4%E6%89%A9%E5%B1%95)

**Failover Cluster**

* 失败自动切换，当出现失败，重试其它服务器。(缺省)
* 通常用于读操作，但重试会带来更长延迟。
* 可通过retries="2"来设置重试次数(不含第一次)。

**Failfast Cluster**

* 快速失败，只发起一次调用，失败立即报错。
* 通常用于非幂等性的写操作，比如新增记录。

**Failsafe Cluster**

* 失败安全，出现异常时，直接忽略。
* 通常用于写入审计日志等操作。

**Failback Cluster**

* 失败自动恢复，后台记录失败请求，定时重发。
* 通常用于消息通知操作。

**Forking Cluster**

* 并行调用多个服务器，只要一个成功即返回。
* 通常用于实时性要求较高的读操作，但需要浪费更多服务资源。
* 可通过forks="2"来设置最大并行数。

**Broadcast Cluster**

* 广播调用所有提供者，逐个调用，任意一台报错则报错。(2.1.0开始支持)
* 通常用于通知所有提供者更新缓存或日志等本地资源信息。

重试次数配置如：(failover集群模式生效)

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:service retries="2" /> |

或：

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:reference retries="2" /> |

或：

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:reference>  <dubbo:method name="findFoo" retries="2" />  </dubbo:reference> |

集群模式配置如：

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:service cluster="failsafe" /> |

或：

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:reference cluster="failsafe" /> |

**负载均衡**

([+](http://code.alibabatech.com/wiki/display/dubbo/Load+Balance-zh)) ([#](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-%E8%B4%9F%E8%BD%BD%E5%9D%87%E8%A1%A1))

|  |  |
| --- | --- |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | 在集群负载均衡时，Dubbo提供了多种均衡策略，缺省为random随机调用。 |

可以自行扩展负载均衡策略，参见：[负载均衡扩展](http://code.alibabatech.com/wiki/display/dubbo/Developer+Guide-zh#DeveloperGuide-zh-%E8%B4%9F%E8%BD%BD%E5%9D%87%E8%A1%A1%E6%89%A9%E5%B1%95)

**Random LoadBalance**

* 随机，按权重设置随机概率。
* 在一个截面上碰撞的概率高，但调用量越大分布越均匀，而且按概率使用权重后也比较均匀，有利于动态调整提供者权重。

**RoundRobin LoadBalance**

* 轮循，按公约后的权重设置轮循比率。
* 存在慢的提供者累积请求问题，比如：第二台机器很慢，但没挂，当请求调到第二台时就卡在那，久而久之，所有请求都卡在调到第二台上。

**LeastActive LoadBalance**

* 最少活跃调用数，相同活跃数的随机，活跃数指调用前后计数差。
* 使慢的提供者收到更少请求，因为越慢的提供者的调用前后计数差会越大。

**ConsistentHash LoadBalance**

* 一致性Hash，相同参数的请求总是发到同一提供者。
* 当某一台提供者挂时，原本发往该提供者的请求，基于虚拟节点，平摊到其它提供者，不会引起剧烈变动。
* 算法参见：<http://en.wikipedia.org/wiki/Consistent_hashing>。
* 缺省只对第一个参数Hash，如果要修改，请配置<dubbo:parameter key="hash.arguments" value="0,1" />
* 缺省用160份虚拟节点，如果要修改，请配置<dubbo:parameter key="hash.nodes" value="320" />

配置如：

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:service interface="..." loadbalance="roundrobin" /> |

或：

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:reference interface="..." loadbalance="roundrobin" /> |

或：

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:service interface="...">  <dubbo:method name="..." loadbalance="roundrobin"/>  </dubbo:service> |

或：

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:reference interface="...">  <dubbo:method name="..." loadbalance="roundrobin"/>  </dubbo:reference> |

**线程模型**

([+](http://code.alibabatech.com/wiki/display/dubbo/Thread+Model-zh)) ([#](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-%E7%BA%BF%E7%A8%8B%E6%A8%A1%E5%9E%8B))



|  |  |
| --- | --- |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | **事件处理线程说明**   * 如果事件处理的逻辑能迅速完成，并且不会发起新的IO请求，比如只是在内存中记个标识，则直接在IO线程上处理更快，因为减少了线程池调度。 * 但如果事件处理逻辑较慢，或者需要发起新的IO请求，比如需要查询数据库，则必须派发到线程池，否则IO线程阻塞，将导致不能接收其它请求。 * 如果用IO线程处理事件，又在事件处理过程中发起新的IO请求，比如在连接事件中发起登录请求，会报“可能引发死锁”异常，但不会真死锁。 |

* Dispatcher
  + all 所有消息都派发到线程池，包括请求，响应，连接事件，断开事件，心跳等。
  + direct 所有消息都不派发到线程池，全部在IO线程上直接执行。
  + message 只有请求响应消息派发到线程池，其它连接断开事件，心跳等消息，直接在IO线程上执行。
  + execution 只请求消息派发到线程池，不含响应，响应和其它连接断开事件，心跳等消息，直接在IO线程上执行。
  + connection 在IO线程上，将连接断开事件放入队列，有序逐个执行，其它消息派发到线程池。
* ThreadPool
  + fixed 固定大小线程池，启动时建立线程，不关闭，一直持有。(缺省)
  + cached 缓存线程池，空闲一分钟自动删除，需要时重建。
  + limited 可伸缩线程池，但池中的线程数只会增长不会收缩。(为避免收缩时突然来了大流量引起的性能问题)。

配置如：

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo |

**协议参考手册**

([+](http://code.alibabatech.com/wiki/display/dubbo/Protocol+Reference-zh)) ([#](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-%E5%8D%8F%E8%AE%AE%E5%8F%82%E8%80%83%E6%89%8B%E5%86%8C))

|  |  |  |
| --- | --- | --- |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | | 推荐使用Dubbo协议 |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | **性能测试报告** 各协议的性能情况，请参见：[性能测试报告](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-%E6%80%A7%E8%83%BD%E6%B5%8B%E8%AF%95%E6%8A%A5%E5%91%8A) ([+](http://code.alibabatech.com/wiki/display/dubbo/Performance+Test+Report-zh)) | |

**dubbo://**

([+](http://code.alibabatech.com/wiki/display/dubbo/Dubbo+Protocol-zh)) ([#](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-dubbo%253A%252F%252F))

|  |  |  |
| --- | --- | --- |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | Dubbo缺省协议采用单一长连接和NIO异步通讯，适合于小数据量大并发的服务调用，以及服务消费者机器数远大于服务提供者机器数的情况。 | |
| http://code.alibabatech.com/wiki/images/icons/emoticons/warning.gif | | Dubbo缺省协议不适合传送大数据量的服务，比如传文件，传视频等，除非请求量很低。 |

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

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

Set default protocol:

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:provider protocol="dubbo" /> |

Set service protocol:

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:service protocol="dubbo" /> |

Multi port:

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:protocol id="dubbo1" name="dubbo" port="20880" />  <dubbo:protocol id="dubbo2" name="dubbo" port="20881" /> |

Dubbo protocol options:

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:protocol name=“dubbo” port=“9090” server=“netty” client=“netty” codec=“dubbo” serialization=“hessian2” charset=“UTF-8” threadpool=“fixed” threads=“100” queues=“0” iothreads=“9” buffer=“8192” accepts=“1000” payload=“8388608” /> |



* Transporter
  + mina, netty, grizzy
* Serialization
  + dubbo, hessian2, java, json
* Dispatcher
  + all, direct, message, execution, connection
* ThreadPool
  + fixed, cached

|  |  |
| --- | --- |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | Dubbo协议缺省每服务每提供者每消费者使用单一长连接，如果数据量较大，可以使用多个连接。 |

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:protocol name="dubbo" connections="2" /> |

* <dubbo:service connections=”0”>或<dubbo:reference connections=”0”>表示该服务使用JVM共享长连接。(缺省)
* <dubbo:service connections=”1”>或<dubbo:reference connections=”1”>表示该服务使用独立长连接。
* <dubbo:service connections=”2”>或<dubbo:reference connections=”2”>表示该服务使用独立两条长连接。

|  |  |
| --- | --- |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | 为防止被大量连接撑挂，可在服务提供方限制大接收连接数，以实现服务提供方自我保护。 |

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:protocol name="dubbo" accepts="1000" /> |

缺省协议，使用基于mina1.1.7+hessian3.2.1的tbremoting交互。

* 连接个数：单连接
* 连接方式：长连接
* 传输协议：TCP
* 传输方式：NIO异步传输
* 序列化：Hessian二进制序列化
* 适用范围：传入传出参数数据包较小（建议小于100K），消费者比提供者个数多，单一消费者无法压满提供者，尽量不要用dubbo协议传输大文件或超大字符串。
* 适用场景：常规远程服务方法调用

为什么要消费者比提供者个数多：  
因dubbo协议采用单一长连接，  
假设网络为千兆网卡(1024Mbit=128MByte)，  
根据测试经验数据每条连接最多只能压满7MByte(不同的环境可能不一样，供参考)，  
理论上1个服务提供者需要20个服务消费者才能压满网卡。

为什么不能传大包：  
因dubbo协议采用单一长连接，  
如果每次请求的数据包大小为500KByte，假设网络为千兆网卡(1024Mbit=128MByte)，每条连接最大7MByte(不同的环境可能不一样，供参考)，  
单个服务提供者的TPS(每秒处理事务数)最大为：128MByte / 500KByte = 262。  
单个消费者调用单个服务提供者的TPS(每秒处理事务数)最大为：7MByte / 500KByte = 14。  
如果能接受，可以考虑使用，否则网络将成为瓶颈。

为什么采用异步单一长连接：  
因为服务的现状大都是服务提供者少，通常只有几台机器，  
而服务的消费者多，可能整个网站都在访问该服务，  
比如Morgan的提供者只有6台提供者，却有上百台消费者，每天有1.5亿次调用，  
如果采用常规的hessian服务，服务提供者很容易就被压跨，  
通过单一连接，保证单一消费者不会压死提供者，  
长连接，减少连接握手验证等，  
并使用异步IO，复用线程池，防止C10K问题。

(1) 约束：

* 参数及返回值需实现Serializable接口
* 参数及返回值不能自定义实现List, Map, Number, Date, Calendar等接口，只能用JDK自带的实现，因为hessian会做特殊处理，自定义实现类中的属性值都会丢失。()
* Hessian序列化，只传成员属性值和值的类型，不传方法或静态变量，兼容情况：(由吴亚军提供)

|  |  |  |
| --- | --- | --- |
| **数据通讯** | **情况** | **结果** |
| A->B | 类A多一种 属性（或者说类B少一种 属性） | 不抛异常，A多的那 个属性的值，B没有， 其他正常 |
| A->B | 枚举A多一种 枚举（或者说B少一种 枚举），A使用多 出来的枚举进行传输 | 抛异常 |
| A->B | 枚举A多一种 枚举（或者说B少一种 枚举），A不使用 多出来的枚举进行传输 | 不抛异常，B正常接 收数据 |
| A->B | A和B的属性 名相同，但类型不相同 | 抛异常 |
| A->B | serialId 不相同 | 正常传输 |

* 总结：会抛异常的情况：枚 举值一边多一种，一边少一种，正好使用了差别的那种，或者属性名相同，类型不同

接口增加方法，对客户端无影响，如果该方法不是客户端需要的，客户端不需要重新部署；  
输入参数和结果集中增加属性，对客户端无影响，如果客户端并不需要新属性，不用重新  
部署；  
输入参数和结果集属性名变化，对客户端序列化无影响，但是如果客户端不重新部署，不管输入还是输出，属性名变化的属性值是获取不到的。  
总结：服务器端和客户端对领域对象并不需要完全一致，而是按照最大匹配原则。

(2) 配置：  
dubbo.properties：

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| dubbo.service.protocol=dubbo |

**rmi://**

([+](http://code.alibabatech.com/wiki/display/dubbo/Rmi+Protocol-zh)) ([#](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-rmi%253A%252F%252F))

|  |  |
| --- | --- |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | RMI协议采用JDK标准的java.rmi.\*实现，采用阻塞式短连接和JDK标准序列化方式。 |

* 如果服务接口继承了java.rmi.Remote接口，可以和原生RMI互操作，即：
  + 提供者用Dubbo的RMI协议暴露服务，消费者直接用标准RMI接口调用，
  + 或者提供方用标准RMI暴露服务，消费方用Dubbo的RMI协议调用。
* 如果服务接口没有继承java.rmi.Remote接口，
  + 缺省Dubbo将自动生成一个com.xxx.XxxService$Remote的接口，并继承java.rmi.Remote接口，并以此接口暴露服务，
  + 但如果设置了<dubbo:protocol name="rmi" codec="spring" />，将不生成$Remote接口，而使用Spring的RmiInvocationHandler接口暴露服务，和Spring兼容。

Define rmi protocol:

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

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

Set default protocol:

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:provider protocol="rmi" /> |

Set service protocol:

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:service protocol="rmi" /> |

Multi port:

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:protocol id="rmi1" name="rmi" port="1099" />  <dubbo:protocol id="rmi2" name="rmi" port="2099" />  <dubbo:service protocol="rmi1" /> |

Spring compatible:

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:protocol name="rmi" codec="spring" /> |

Java标准的远程调用协议。

* 连接个数：多连接
* 连接方式：短连接
* 传输协议：TCP
* 传输方式：同步传输
* 序列化：Java标准二进制序列化
* 适用范围：传入传出参数数据包大小混合，消费者与提供者个数差不多，可传文件。
* 适用场景：常规远程服务方法调用，与原生RMI服务互操作

(1) 约束：

* 参数及返回值需实现Serializable接口
* dubbo配置中的超时时间对rmi无效，需使用java启动参数设置：-Dsun.rmi.transport.tcp.responseTimeout=3000，参见下面的RMI配置。

(2) 配置：  
dubbo.properties：

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| dubbo.service.protocol=rmi |

(3) RMI配置：

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| java -Dsun.rmi.transport.tcp.responseTimeout=3000 |

更多RMI优化参数请查看：  
<http://download.oracle.com/docs/cd/E17409_01/javase/6/docs/technotes/guides/rmi/sunrmiproperties.html>

**hessian://**

([+](http://code.alibabatech.com/wiki/display/dubbo/Hessian+Protocol-zh)) ([#](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-hessian%253A%252F%252F))

|  |  |
| --- | --- |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | Hessian协议用于集成Hessian的服务，Hessian底层采用Http通讯，采用Servlet暴露服务，Dubbo缺省内嵌Jetty作为服务器实现。 |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | Hessian是Caucho开源的一个RPC框架：<http://hessian.caucho.com>，其通讯效率高于WebService和Java自带的序列化。 |

依赖：

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dependency>  <groupId>com.caucho</groupId>  <artifactId>hessian</artifactId>  <version>4.0.7</version>  </dependency> |

可以和原生Hessian服务互操作，即：

* 提供者用Dubbo的Hessian协议暴露服务，消费者直接用标准Hessian接口调用，
* 或者提供方用标准Hessian暴露服务，消费方用Dubbo的Hessian协议调用。

基于Hessian的远程调用协议。

* 连接个数：多连接
* 连接方式：短连接
* 传输协议：HTTP
* 传输方式：同步传输
* 序列化：Hessian二进制序列化
* 适用范围：传入传出参数数据包较大，提供者比消费者个数多，提供者压力较大，可传文件。
* 适用场景：页面传输，文件传输，或与原生hessian服务互操作

(1) 约束：

* 参数及返回值需实现Serializable接口
* 参数及返回值不能自定义实现List, Map, Number, Date, Calendar等接口，只能用JDK自带的实现，因为hessian会做特殊处理，自定义实现类中的属性值都会丢失。

(2) 配置：  
Define hessian protocol:

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:protocol name="hessian" port="8080" server="jetty" /> |

Set default protocol:

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:provider protocol="hessian" /> |

Set service protocol:

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:service protocol="hessian" /> |

Multi port:

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:protocol id="hessian1" name="hessian" port="8080" />  <dubbo:protocol id="hessian2" name="hessian" port="8081" /> |

Directly provider:

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:reference id="helloService" interface="HelloWorld" url="hessian://10.20.153.10:8080/helloWorld" /> |

**h4. Jetty Server: (default)**

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:protocol ... server="jetty" /> |

**h4. Servlet Bridge Server: (recommend)**

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:protocol ... server="servlet" /> |

web.xml：

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <servlet>  <servlet-name>dubbo</servlet-name>  <servlet-class>com.alibaba.dubbo.remoting.http.servlet.DispatcherServlet</servlet-class>  <load-on-startup>1</load-on-startup>  </servlet>  <servlet-mapping>  <servlet-name>dubbo</servlet-name>  <url-pattern>/\*</url-pattern>  </servlet-mapping> |

注意，如果使用servlet派发请求：

* 协议的端口<dubbo:protocol port="8080" />必须与servlet容器的端口相同，
* 协议的上下文路径<dubbo:protocol contextpath="foo" />必须与servlet应用的上下文路径相同。

**http://**

([+](http://code.alibabatech.com/wiki/display/dubbo/Http+Protocol-zh)) ([#](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-http%253A%252F%252F))

|  |  |  |
| --- | --- | --- |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | 采用Spring的HttpInvoker实现 | |
| http://code.alibabatech.com/wiki/images/icons/emoticons/warning.gif | | 2.3.0以上版本支持 |

基于http表单的远程调用协议。参见：[HTTP协议使用说明]

* 连接个数：多连接
* 连接方式：短连接
* 传输协议：HTTP
* 传输方式：同步传输
* 序列化：表单序列化
* 适用范围：传入传出参数数据包大小混合，提供者比消费者个数多，可用浏览器查看，可用表单或URL传入参数，暂不支持传文件。
* 适用场景：需同时给应用程序和浏览器JS使用的服务。

(1) 约束：

* 参数及返回值需符合Bean规范

(2) 配置：  
dubbo.xml：

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:protocol name="http" port="8080" /> |

**h4. Jetty Server: (default)**

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:protocol ... server="jetty" /> |

**h4. Servlet Bridge Server: (recommend)**

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:protocol ... server="servlet" /> |

web.xml：

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <servlet>  <servlet-name>dubbo</servlet-name>  <servlet-class>com.alibaba.dubbo.remoting.http.servlet.DispatcherServlet</servlet-class>  <load-on-startup>1</load-on-startup>  </servlet>  <servlet-mapping>  <servlet-name>dubbo</servlet-name>  <url-pattern>/\*</url-pattern>  </servlet-mapping> |

注意，如果使用servlet派发请求：

* 协议的端口<dubbo:protocol port="8080" />必须与servlet容器的端口相同，
* 协议的上下文路径<dubbo:protocol contextpath="foo" />必须与servlet应用的上下文路径相同。

**webservice://**

([+](http://code.alibabatech.com/wiki/display/dubbo/WebService+Protocol)) ([#](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-webservice%253A%252F%252F))

|  |  |  |
| --- | --- | --- |
| http://code.alibabatech.com/wiki/images/icons/emoticons/warning.gif | | 2.3.0以上版本支持。 |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | 基于CXF的[frontend-simple](http://search.maven.org/#search%7Cga%7C1%7Ca%3A%22cxf-rt-frontend-simple%22)和[transports-http](http://search.maven.org/#search%7Cga%7C1%7Ca%3A%22cxf-rt-transports-http%22)实现。 | |

|  |  |
| --- | --- |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | CXF是Apache开源的一个RPC框架：<http://cxf.apache.org>，由Xfire和Celtix合并而来 。 |

依赖：

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dependency>  <groupId>org.apache.cxf</groupId>  <artifactId>cxf-rt-frontend-simple</artifactId>  <version>2.6.1</version>  </dependency>  <dependency>  <groupId>org.apache.cxf</groupId>  <artifactId>cxf-rt-transports-http</artifactId>  <version>2.6.1</version>  </dependency> |

可以和原生WebService服务互操作，即：

* 提供者用Dubbo的WebService协议暴露服务，消费者直接用标准WebService接口调用，
* 或者提供方用标准WebService暴露服务，消费方用Dubbo的WebService协议调用。

基于WebService的远程调用协议。

* 连接个数：多连接
* 连接方式：短连接
* 传输协议：HTTP
* 传输方式：同步传输
* 序列化：SOAP文本序列化
* 适用场景：系统集成，跨语言调用。

(1) 约束：

* 参数及返回值需实现Serializable接口
* 参数尽量使用基本类型和POJO。

(2) 配置：  
Define hessian protocol:

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:protocol name="webservice" port="8080" server="jetty" /> |

Set default protocol:

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:provider protocol="webservice" /> |

Set service protocol:

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:service protocol="webservice" /> |

Multi port:

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:protocol id="webservice1" name="webservice" port="8080" />  <dubbo:protocol id="webservice2" name="webservice" port="8081" /> |

Directly provider:

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:reference id="helloService" interface="HelloWorld" url="webservice://10.20.153.10:8080/com.foo.HelloWorld" /> |

WSDL:

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <http://10.20.153.10:8080/com.foo.HelloWorld?wsdl> |

**h4. Jetty Server: (default)**

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:protocol ... server="jetty" /> |

**h4. Servlet Bridge Server: (recommend)**

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:protocol ... server="servlet" /> |

web.xml：

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <servlet>  <servlet-name>dubbo</servlet-name>  <servlet-class>com.alibaba.dubbo.remoting.http.servlet.DispatcherServlet</servlet-class>  <load-on-startup>1</load-on-startup>  </servlet>  <servlet-mapping>  <servlet-name>dubbo</servlet-name>  <url-pattern>/\*</url-pattern>  </servlet-mapping> |

注意，如果使用servlet派发请求：

* 协议的端口<dubbo:protocol port="8080" />必须与servlet容器的端口相同，
* 协议的上下文路径<dubbo:protocol contextpath="foo" />必须与servlet应用的上下文路径相同。

**thrift://**

([+](http://code.alibabatech.com/wiki/display/dubbo/Thrift+Protocol)) ([#](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-thrift%253A%252F%252F))

|  |  |  |
| --- | --- | --- |
| http://code.alibabatech.com/wiki/images/icons/emoticons/warning.gif | | 2.3.0以上版本支持。 |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | **Thrift说明** Thrift是Facebook捐给Apache的一个RPC框架，参见：<http://thrift.apache.org> | |

|  |  |
| --- | --- |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | **dubbo thrift协议** 当前 dubbo 支持的 thrift 协议是对 thrift 原生协议的扩展，在原生协议的基础上添加了一些额外的头信息，比如service name，magic number等。使用dubbo thrift协议同样需要使用thrift的idl compiler编译生成相应的java代码，后续版本中会在这方面做一些增强。 |

示例：<https://github.com/alibaba/dubbo/tree/master/dubbo-rpc/dubbo-rpc-thrift/src/test/java/com/alibaba/dubbo/rpc/protocol/thrift/examples>

依赖：

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dependency>  <groupId>org.apache.thrift</groupId>  <artifactId>libthrift</artifactId>  <version>0.8.0</version>  </dependency> |

所有服务共用一个端口：(与原生Thrift不兼容)

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:protocol name="thrift" port="3030" /> |

Thrift不支持数据类型：

* null值 (不能在协议中传递null值)

**memcached://**

([+](http://code.alibabatech.com/wiki/display/dubbo/Memcached+Protocol)) ([#](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-memcached%253A%252F%252F))

|  |  |  |
| --- | --- | --- |
| http://code.alibabatech.com/wiki/images/icons/emoticons/warning.gif | | 2.3.0以上版本支持。 |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | **Memcached说明** Memcached是一个高效的KV缓存服务器，参见：<http://memcached.org/> | |

可以通过脚本或监控中心手工填写表单注册memcached服务的地址：

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| RegistryFactory registryFactory = ExtensionLoader.getExtensionLoader(RegistryFactory.class).getAdaptiveExtension();  Registry registry = registryFactory.getRegistry(URL.valueOf("zookeeper://10.20.153.10:2181"));  registry.register(URL.valueOf("memcached://10.20.153.11/com.foo.BarService?category=providers&dynamic=false&application=foo&group=member&loadbalance=consistenthash")); |

然后在客户端使用时，不需要感知Memcached的地址：

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:reference id="cache" interface="java.util.Map" group="member" /> |

或者，点对点直连：

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:reference id="cache" interface="java.util.Map" url="memcached://10.20.153.10:11211" /> |

也可以使用自定义接口：

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:reference id="cache" interface="com.foo.CacheService" url="memcached://10.20.153.10:11211" /> |

方法名建议和memcached的标准方法名相同，即：get(key), set(key, value), delete(key)。

如果方法名和memcached的标准方法名不相同，则需要配置映射关系：(其中"p:xxx"为spring的标准p标签)

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:reference id="cache" interface="com.foo.CacheService" url="memcached://10.20.153.10:11211" p:set="putFoo" p:get="getFoo" p:delete="removeFoo" /> |

**redis://**

([+](http://code.alibabatech.com/wiki/display/dubbo/Redis+Protocol)) ([#](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-redis%253A%252F%252F))

|  |  |  |
| --- | --- | --- |
| http://code.alibabatech.com/wiki/images/icons/emoticons/warning.gif | | 2.3.0以上版本支持。 |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | **Redis说明** Redis是一个高效的KV存储服务器，参见：<http://redis.io> | |

可以通过脚本或监控中心手工填写表单注册redis服务的地址：

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| RegistryFactory registryFactory = ExtensionLoader.getExtensionLoader(RegistryFactory.class).getAdaptiveExtension();  Registry registry = registryFactory.getRegistry(URL.valueOf("zookeeper://10.20.153.10:2181"));  registry.register(URL.valueOf("redis://10.20.153.11/com.foo.BarService?category=providers&dynamic=false&application=foo&group=member&loadbalance=consistenthash")); |

然后在客户端使用时，不需要感知Redis的地址：

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:reference id="store" interface="java.util.Map" group="member" /> |

或者，点对点直连：

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:reference id="store" interface="java.util.Map" url="redis://10.20.153.10:6379" /> |

也可以使用自定义接口：

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:reference id="store" interface="com.foo.StoreService" url="redis://10.20.153.10:6379" /> |

方法名建议和redis的标准方法名相同，即：get(key), set(key, value), delete(key)。

如果方法名和redis的标准方法名不相同，则需要配置映射关系：(其中"p:xxx"为spring的标准p标签)

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <dubbo:reference id="cache" interface="com.foo.CacheService" url="memcached://10.20.153.10:11211" p:set="putFoo" p:get="getFoo" p:delete="removeFoo" /> |

**快速启动**

([+](http://code.alibabatech.com/wiki/display/dubbo/Quick+Start-zh)) ([#](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-%E5%BF%AB%E9%80%9F%E5%90%AF%E5%8A%A8))

|  |  |
| --- | --- |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | Dubbo采用全Spring配置方式，透明化接入应用，对应用没有任何API侵入，只需用Spring加载Dubbo的配置即可，Dubbo基于Spring的Schema扩展进行加载。 |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | 如果不想使用Spring配置，而希望通过API的方式进行调用（不推荐），请参见：[API配置](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-API%E9%85%8D%E7%BD%AE) ([+](http://code.alibabatech.com/wiki/display/dubbo/API+Config-zh)) |

**服务提供者**

([#](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-%E6%9C%8D%E5%8A%A1%E6%8F%90%E4%BE%9B%E8%80%85))

|  |  |
| --- | --- |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | 完整安装步骤，请参见：[示例提供者安装](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh#AdministratorGuide-zh-%E7%A4%BA%E4%BE%8B%E6%8F%90%E4%BE%9B%E8%80%85%E5%AE%89%E8%A3%85) ([+](http://code.alibabatech.com/wiki/display/dubbo/Demo+Provider+Installation-zh)) |

定义服务接口: (该接口需单独打包，在服务提供方和消费方共享)

**DemoService.java**

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| package com.alibaba.dubbo.demo;  public interface DemoService {  String sayHello(String name);  } |

在服务提供方实现接口：(对服务消费方隐藏实现)

**DemoServiceImpl.java**

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| package com.alibaba.dubbo.demo.provider;  import com.alibaba.dubbo.demo.DemoService;  public class DemoServiceImpl implements DemoService {  public String sayHello(String name) {  return "Hello " + name;  }  } |

用Spring配置声明暴露服务：

**provider.xml**

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <beans xmlns="[http://www.springframework.org/schema/beans"](http://www.springframework.org/schema/beans)  xmlns:xsi="[http://www.w3.org/2001/XMLSchema-instance"](http://www.w3.org/2001/XMLSchema-instance)  xmlns:dubbo="[http://code.alibabatech.com/schema/dubbo"](http://code.alibabatech.com/schema/dubbo)  xsi:schemaLocation="http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans.xsd  http://code.alibabatech.com/schema/dubbo  http://code.alibabatech.com/schema/dubbo/dubbo.xsd  ">  <!-- 提供方应用信息，用于计算依赖关系 -->  <dubbo:application name="hello-world-app" />  <!-- 使用multicast广播注册中心暴露服务地址 -->  <dubbo:registry address="multicast://224.5.6.7:1234" />  <!-- 使用zookeeper注册中心暴露发现服务地址 -->  <dubbo:registry protocol="zookeeper" address="172.16.103.146:2181,172.16.103.147:2181" />  <!-- 用dubbo协议在20880端口暴露服务 -->  <dubbo:protocol name="dubbo" port="20880" />  <!-- 声明需要暴露的服务接口 -->  <dubbo:service interface="com.alibaba.dubbo.demo.DemoService" ref="demoService" />  <!-- 和本地bean一样实现服务 -->  <bean id="demoService" class="com.alibaba.dubbo.demo.provider.DemoServiceImpl" />  </beans> |

加载Spring配置：

**Provider.java**

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| import org.springframework.context.support.ClassPathXmlApplicationContext;  public class Provider {  public static void main(String[] args) throws Exception {  ClassPathXmlApplicationContext context = new ClassPathXmlApplicationContext(new String[] {"provider.xml"});  context.start();  System.in.read(); // 按任意键退出  }  } |

**服务消费者**

([#](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh#UserGuide-zh-%E6%9C%8D%E5%8A%A1%E6%B6%88%E8%B4%B9%E8%80%85))

|  |  |
| --- | --- |
| http://code.alibabatech.com/wiki/images/icons/emoticons/check.gif | 完整安装步骤，请参见：[示例消费者安装](http://code.alibabatech.com/wiki/display/dubbo/Administrator+Guide-zh#AdministratorGuide-zh-%E7%A4%BA%E4%BE%8B%E6%B6%88%E8%B4%B9%E8%80%85%E5%AE%89%E8%A3%85) ([+](http://code.alibabatech.com/wiki/display/dubbo/Demo+Consumer+Installation-zh)) |

通过Spring配置引用远程服务：

**consumer.xml**

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?>  <beans xmlns="[http://www.springframework.org/schema/beans"](http://www.springframework.org/schema/beans)  xmlns:xsi="[http://www.w3.org/2001/XMLSchema-instance"](http://www.w3.org/2001/XMLSchema-instance)  xmlns:dubbo="[http://code.alibabatech.com/schema/dubbo"](http://code.alibabatech.com/schema/dubbo)  xsi:schemaLocation="http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans.xsd  http://code.alibabatech.com/schema/dubbo  http://code.alibabatech.com/schema/dubbo/dubbo.xsd  ">  <!-- 消费方应用名，用于计算依赖关系，不是匹配条件，不要与提供方一样 -->  <dubbo:application name="consumer-of-helloworld-app" />  <!-- 使用multicast广播注册中心暴露发现服务地址 -->  <dubbo:registry address="multicast://224.5.6.7:1234" />  <!-- 使用zookeeper注册中心暴露发现服务地址 -->  <dubbo:registry protocol="zookeeper" address="172.16.103.146:2181,172.16.103.147:2181" />  <!-- 生成远程服务代理，可以和本地bean一样使用demoService -->  <dubbo:reference id="demoService" interface="com.alibaba.dubbo.demo.DemoService" />  </beans> |

加载Spring配置，并调用远程服务：(也可以使用IoC注入)

**Consumer.java**

[?](http://code.alibabatech.com/wiki/display/dubbo/User+Guide-zh)

|  |
| --- |
| import org.springframework.context.support.ClassPathXmlApplicationContext;  import com.alibaba.dubbo.demo.DemoService;  public class Consumer {  public static void main(String[] args) throws Exception {  ClassPathXmlApplicationContext context = new ClassPathXmlApplicationContext(new String[] {"consumer.xml"});  context.start();  DemoService demoService = (DemoService)context.getBean("demoService"); // 获取远程服务代理  String hello = demoService.sayHello("world"); // 执行远程方法  System.out.println( hello ); // 显示调用结果  }  } |

**基于上品现有系统的迁移**

**创建订单接口改造**

orderController 🡪 createOrder (HttpServletRequest request,

HttpServletResponse response)

内部调用接口

* 1. dpOrderService/isAllowBuyPresale 远程调用O2DP
  2. getActiveProduct(List<OrderDetailVO> orderVoDetailList,OrderVO orderVO) 调用本地方法

2.1 dpOrderService/getProDetailInfor远程调用O2DP

* 1. createOrder(orderVo) 调用本地service 接口

3.1 dpOrderService/getNextOrderNumber远程调用O2DP

3.2 dpOrderService/getProDetailInfor 远程调用O2DP

3.3 dpOrderService/getProDetailInfoByProDetailSid远程调用O2DP

3.4 dpOrderService/operStock远程调用O2DP

* 1. dpOrderService/saveOrdersInfor 远程调用O2DP

4.1 orderService/queryOmsOrder 远程调用OMS

**相关应用jar**



**ShopinO2DP 修改（服务端）**

1．添加 接口 并对一下接口实现

**public** **interface** NumberGeneraterDService {

**public** String getNextOrderNumber();

**public** String getNextSaleNumber();

**public** String getNextRefundsNumber();

}

**public** **interface** OrderDService {

**public** String getProDetailInfor(String detaillist);

**public** String getProDetailInfoByProDetailSid(String proDetailSid);

**public** String operStock(String operStock);

**public** String saveOrdersInfor(String orderNo);

}

**public** **interface** PresaleOrdersDService {

**public** **boolean** isAllowBuyPresale(String trailDetailSid);

}

2．添加applicationContext-dubbo.xml 配置文件

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:aop="http://www.springframework.org/schema/aop"

xmlns:dubbo="http://code.alibabatech.com/schema/dubbo"

xmlns:context="http://www.springframework.org/schema/context"

xmlns:task="http://www.springframework.org/schema/task" xmlns:tx="http://www.springframework.org/schema/tx"

xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.0.xsd

http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-3.0.xsd

http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.0.xsd

http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-3.0.xsd

http://www.springframework.org/schema/task http://www.springframework.org/schema/task/spring-task-3.0.xsd

http://code.alibabatech.com/schema/dubbo http://code.alibabatech.com/schema/dubbo/dubbo.xsd">

<description>Spring-dubbo公共配置文件</description>

<!-- 提供方应用信息，用于计算依赖关系 -->

<dubbo:application name="o2dpdubbo" />

<!-- 使用zookeeper注册中心暴露服务地址 -->

<dubbo:registry protocol="zookeeper" address="172.16.103.147:2181,172.16.103.146:2181" />

<!-- 用dubbo协议在20880端口暴露服务 -->

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

<!-- 声明需要暴露的服务接口 -->

<dubbo:service interface="net.shopin.order.o2dp.dubbo.service.NumberGeneraterDService" ref="numberGeneraterDService" />

<dubbo:service interface="net.shopin.order.o2dp.dubbo.service.OrderDService" ref="orderDService" />

<dubbo:service interface="net.shopin.order.o2dp.dubbo.service.PresaleOrdersDService" ref="presaleOrdersDService" />

<dubbo:consumer check="false" />

<!-- 生成远程服务代理，可以和本地bean一样使用demoService -->

<dubbo:reference id="orderOmsDService" interface="net.shopin.order.oms.dubbo.service.OrderOmsDService" />

</beans>

**ShopinOMS 修改（消费端）**

1．添加需要调用的接口 无需实现

**public** **interface** NumberGeneraterDService {

**public** String getNextOrderNumber();

**public** String getNextSaleNumber();

**public** String getNextRefundsNumber();

}

**public** **interface** OrderDService {

**public** String getProDetailInfor(String detaillist);

**public** String getProDetailInfoByProDetailSid(String proDetailSid);

**public** String operStock(String operStock);

**public** String saveOrdersInfor(String orderNo);

}

**public** **interface** PresaleOrdersDService {

**public** **boolean** isAllowBuyPresale(String trailDetailSid);

}

2．添加 applicationContext-dubbo.xml 配置文件

<?xml version="1.0" encoding="UTF-8"?>

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="[ttp://www.w3.org/2001/XMLSchema-instance"](ttp://www.w3.org/2001/XMLSchema-instance%22)xmlns:aop="http://www.springframework.org/schema/aop"

xmlns:dubbo="http://code.alibabatech.com/schema/dubbo"

xmlns:context="http://www.springframework.org/schema/context"

xmlns:task=<http://www.springframework.org/schema/task> xmlns:tx="http://www.springframework.org/schema/tx"

xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-3.0.xsd

http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-3.0.xsd

http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.0.xsd

http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-3.0.xsd

http://www.springframework.org/schema/task http://www.springframework.org/schema/task/spring-task-3.0.xsd

http://code.alibabatech.com/schema/dubbo http://code.alibabatech.com/schema/dubbo/dubbo.xsd">

<description>Spring-dubbo公共配置文件</description>

<!-- 提供方应用信息，用于计算依赖关系 -->

<dubbo:application name="omsdubbo" />

<!-- 使用zookeeper注册中心暴露服务地址 -->

<dubbo:registry protocol="zookeeper" address="172.16.103.146:2181,172.16.103.147:2181" />

<!-- 用dubbo协议在20890端口暴露服务 -->

<dubbo:protocol name="dubbo" port="20890" />

<!-- 声明需要暴露的服务接口 -->

<dubbo:service interface="net.shopin.order.oms.dubbo.service.OrderOmsDService" ref="orderOmsDService" />

<dubbo:consumer check="false" />

<!-- 生成远程服务代理，可以和本地bean一样使用demoService -->

<dubbo:reference id="numberGeneraterDService" interface="net.shopin.order.o2dp.dubbo.service.NumberGeneraterDService" />

<dubbo:reference id="orderDService" interface="net.shopin.order.o2dp.dubbo.service.OrderDService" />

<dubbo:reference id="presaleOrdersDService" interface="net.shopin.order.o2dp.dubbo.service.PresaleOrdersDService" />

</beans>

3．修改调用的代码

在public class OrderController {} 中首先需要

@Autowired//注入远程调用的方法

**private** OrderDService orderDService = **null**;

在调用时直接同本地方法一下调用

String s=orderDService.saveOrdersInfor(omsOrder.getOrderNo());

**上品目前部署 生产环境**

**Zookeeper 部署集群**

节点1 192.168.200.133  
节点2 192.168.200.134  
节点3 192.168.200.136

部署位置 /usr/local/zookeeper-3.3.6/

启动zookeeper 进入bin目录下

启动 ./zkServer start

停止 ./zkServer stop

查看状态 ./zkServer status

**Dubbo 服务管理界面**

应用连接地址 <http://192.168.200.133>

部署位置 /usr/local/apache-tomcat-keeper-monitor

启动服务 进入bin目录下

启动 ./startup.sh

停止 ps -ef|grep apache-tomcat-keeper-monitor 获取进程号 kill -9 [进程号]

查看日志 tail –f ./usr/local/apache-tomcat-keeper-monitor/logs/catalina.out

**Dubbo 监控界面**

应用连接地址 http://192.168.200.133:8580/

部署位置 /usr/local/dubbo-monitor-simple-2.5.3

启动服务 进入bin目录下

启动 ./start.sh

停止 ./stop.sh

**上品目前部署 开发环境**

**Zookeeper 部署集群**

节点1 172.16.103.146

节点2 172.16.103.147

节点3 172.16.103.145

部署位置 /usr/local/zookeeper-3.3.6/

启动zookeeper 进入bin目录下

启动 ./zkServer start

停止 ./zkServer stop

查看状态 ./zkServer status

**Dubbo 服务管理界面**

应用连接地址

<http://172.16.103.147>

部署位置 /usr/local/apache-tomcat-keeper-monitor

启动服务 进入bin目录下

启动 ./startup.sh

停止 ps -ef|grep apache-tomcat-keeper-monitor 获取进程号 kill -9 [进程号]

查看日志 tail –f ./usr/local/apache-tomcat-keeper-monitor/logs/catalina.out

**Dubbo 监控界面**

应用连接地址 <http://172.16.103.147:8580/> 目前监控界面没有开启，暂定状态

部署位置 /usr/local/dubbo-monitor-simple-2.5.3

启动服务 进入bin目录下

启动 ./start.sh

停止 ./stop.sh

**上品目前部署 测试环境**

**Zookeeper 部署单节点**

节点1 172.16.103.130 单节点

部署位置 /usr/local/zookeeper-3.3.6/

启动zookeeper 进入bin目录下

启动 ./zkServer start

停止 ./zkServer stop

查看状态 ./zkServer status

**Dubbo 服务管理界面**

应用连接地址 <http://172.16.103.130> 暂定

部署位置 /usr/local/apache-tomcat-keeper-monitor

启动服务 进入bin目录下

启动 ./startup.sh

停止 ps -ef|grep apache-tomcat-keeper-monitor 获取进程号 kill -9 [进程号]

查看日志 tail –f ./usr/local/apache-tomcat-keeper-monitor/logs/catalina.out

**Dubbo 监控界面**

应用连接地址 [http://172.16.103.130:8580/](http://172.16.103.147:8580/) 目前监控界面没有开启，暂定状态

部署位置 /usr/local/dubbo-monitor-simple-2.5.3

启动服务 进入bin目录下

启动 ./start.sh

停止 ./stop.sh