
Table of Contents

Introduction	1.1
1 安装手册	1.2
1.1 示例提供者安装	1.2.1
1.2 示例消费者安装	1.2.2
1.3 Zookeeper注册中心安装	1.2.3
1.4 Redis注册中心安装	1.2.4
1.5 Simple注册中心安装	1.2.5
1.6 Simple监控中心安装	1.2.6
1.7 管理控制台安装	1.2.7
2 运维手册	1.3
2.1 管理控制台运维	1.3.1

✔ 推荐使用Zookeeper注册中心

- 你可以只运行 `Demo Provider` 和 `Demo Consumer`，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` 和 `Demo Consumer`，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` 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` 后，修改 `Demo Provider`，`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`，值改为 `redis://127.0.0.1:6379`，如果换成 `Simple Registry`，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:

```
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:

```
vi conf/dubbo.properties
```

- 启动:

```
./bin/start.sh
```

- 停止:

```
./bin/stop.sh
```

- 重启:

```
./bin/restart.sh
```

- 调试:

```
./bin/start.sh debug
```

- 系统状态:

```
./bin/dump.sh
```

- 总控入口:

```
./bin/server.sh start  
./bin/server.sh stop  
./bin/server.sh restart  
./bin/server.sh debug  
./bin/server.sh dump
```

- 标准输出:

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

- 命令行: (See: [Telnet Command Reference](#))

```
telnet 127.0.0.1 20880  
help
```

Or:

```
echo status | nc -i 1 127.0.0.1 20880
```

- 安装:

```
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:

```
vi conf/dubbo.properties
```

- 启动:

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

- 停止:

```
./bin/stop.sh
```

- 重启:

```
./bin/restart.sh
```

- 调试:

```
./bin/start.sh debug
```

- 系统状态:

```
./bin/dump.sh
```

- 总控入口:

```
./bin/server.sh start  
./bin/server.sh stop  
./bin/server.sh restart  
./bin/server.sh debug  
./bin/server.sh dump
```

- 标准输出:

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

- ✔ 建议使用dubbo-2.3.3以上版本的zookeeper注册中心客户端
- ✔ Zookeeper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。
- ✔ 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。
- ✔ 如果需要，可以考虑使用taobao的zookeeper监控：<http://rdc.taobao.com/team/jm/archives/1450>

● 安装:

```
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:

```
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

```
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

```
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)

```
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
```

```
1
```

- 启动:

```
./bin/zkServer.sh start
```

- 停止:

```
./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:

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

Or:

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

✅Redis说明

Redis是一个高效的KV存储服务器，参见：<http://redis.io>

✅Redis使用

使用方式参见: [Redis使用手册](#)，只需搭一个原生的Redis服务器，并将[Quick Start](#)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

✅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。

⚠️2.1.0以上版本支持

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

- 安装:

```
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:

```
vi redis.conf
```

- 启动:

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

- 停止:

```
killall redis-server
```

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

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

或者：

```
telnet 127.0.0.1 6379  
hgetall /dubbo/com.foo.BarService/providers
```

⚠️ Simple Registry 没有经过严格测试，可能不健壮，并且不支持集群，不建议用于生产环境。

- 安装:

```
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:

```
vi conf/dubbo.properties
```

- 启动:

```
./bin/start.sh
```

- 停止:

```
./bin/stop.sh
```

- 重启:

```
./bin/restart.sh
```

- 调试:

```
./bin/start.sh debug
```

- 系统状态:

```
./bin/dump.sh
```

- 总控入口:

```
./bin/server.sh start
./bin/server.sh stop
./bin/server.sh restart
./bin/server.sh debug
./bin/server.sh dump
```

- 标准输出:

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

- 命令行: (See: [Telnet Command Reference](#))

```
telnet 127.0.0.1 9090
help
```

Or:

```
echo status | nc -i 1 127.0.0.1 9090
```

✔ 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。

⚠ 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共享磁盘。

⚠ charts目录必须放在jetty.directory下，否则页面上访问不了。

- 安装:

```
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:

```
vi conf/dubbo.properties
```

- 启动:

```
./bin/start.sh
```

- 停止:

```
./bin/stop.sh
```

- 重启:

```
./bin/restart.sh
```

- 调试:

```
./bin/start.sh debug
```

- 系统状态:

```
./bin/dump.sh
```

- 总控入口:

```
./bin/server.sh start  
./bin/server.sh stop  
./bin/server.sh restart  
./bin/server.sh debug  
./bin/server.sh dump
```

- 标准输出:

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

- 命令行: (See: [Telnet Command Reference](#))

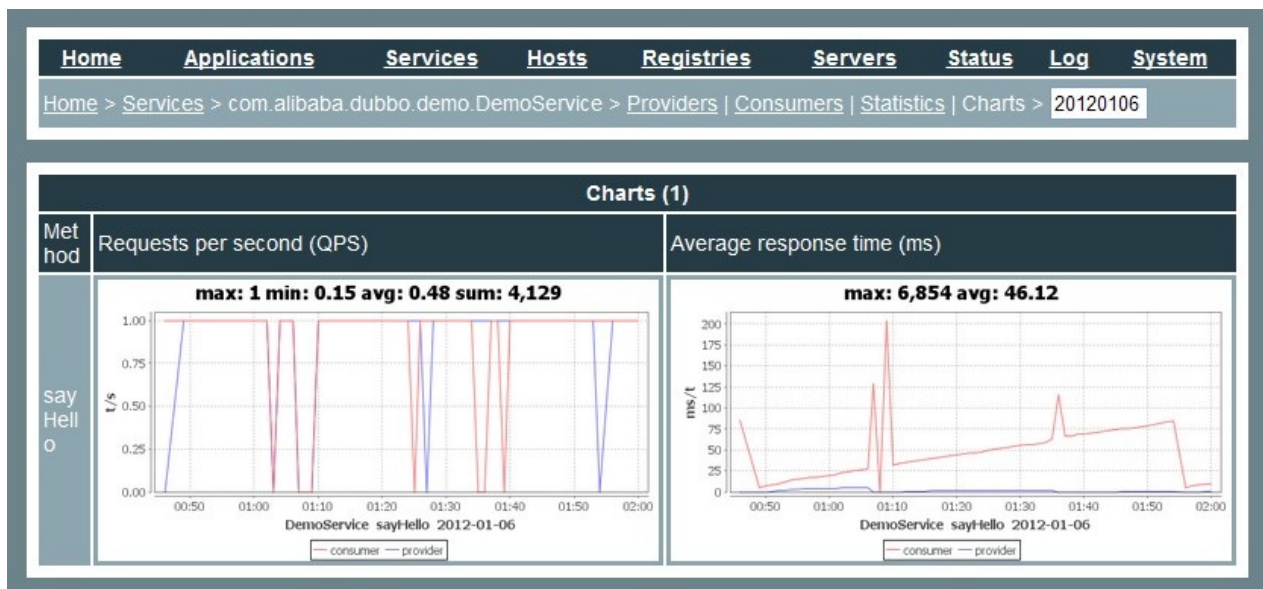
```
telnet 127.0.0.1 7070  
help
```

Or:

```
echo status | nc -i 1 127.0.0.1 7070
```

访问:

```
http://127.0.0.1:8080
```



✔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:

```
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  
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)

```
vi webapps/ROOT/WEB-INF/dubbo.properties  
dubbo.properties  
dubbo.registry.address=zookeeper://127.0.0.1:2181  
dubbo.admin.root.password=root  
dubbo.admin.guest.password=guest
```

- 启动:

```
./bin/startup.sh
```

- 停止:

```
./bin/shutdown.sh
```

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

```
http://127.0.0.1:8080/
```


管理控制台运维

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

2.1 管理控制台运维



服务应用页面



添加路由规则页面

首页

服务治理

系统管理

帮助

新增 路由规则

路由规则

新增

服务名 | 应用名 | 机器IP

SEARCH com.alibaba.dubbo.demo.DemoService

提供者

消费者

应用

路由规则

动态配置

访问控制

权重调节

负载均衡

负责人

返回

路由名称: 可中文, 由1-200个字符组成

优先级: 0 数字越大越优先

服务名: com.alibaba.dubbo.demo.DemoService

方法名: 请选择 只有Dubbo2.0.0以上版本的服务消费端支持按方法路由, 多个方法名用逗号分隔

匹配条件 匹配 不匹配 当消费者满足匹配条件时使用当前规则进行过滤

消费者IP地址: 多个值用逗号分隔, 以星号结尾表示通配地址段

消费者应用名: 多个值用逗号分隔

消费者集群: 可通过菜单"服务控制"->"服务器集群"管理

过滤规则 匹配 不匹配 满足过滤规则的提供者地址将被推送给消费者

提供者IP地址: 多个值用逗号分隔, 以星号结尾表示通配地址段

提供者集群: 可通过菜单"服务控制"->"服务器集群"管理

提供者协议:

提供者端口:

保存

添加动态配置页面

首页

服务治理

系统管理

帮助

新增 动态配置

动态配置

新增

服务名 | 应用名 | 机器IP

SEARCH com.alibaba.dubbo.demo.DemoService

提供者

消费者

应用

路由规则

动态配置

访问控制

权重调节

负载均衡

负责人

返回

服务名: com.alibaba.dubbo.demo.DemoService

消费者应用名:

只推送给指定消费者地址: 不填表示对消费者应用的所有机器生效

状态: 禁用

动态配置

参数名: 参数值: 方法级配置如: findPerson.timeout=1000

新增参数

服务降级

所有方法的Mock值: 容错 示例: return null/empty/JSON或throw com.foo.BarException

新增方法

保存

服务注册

服务降级

路由规则

访问控制

动态配置

权重调节

负载均衡

服务负责人