

08-Dubbo

- 1, Dubbo简介

-

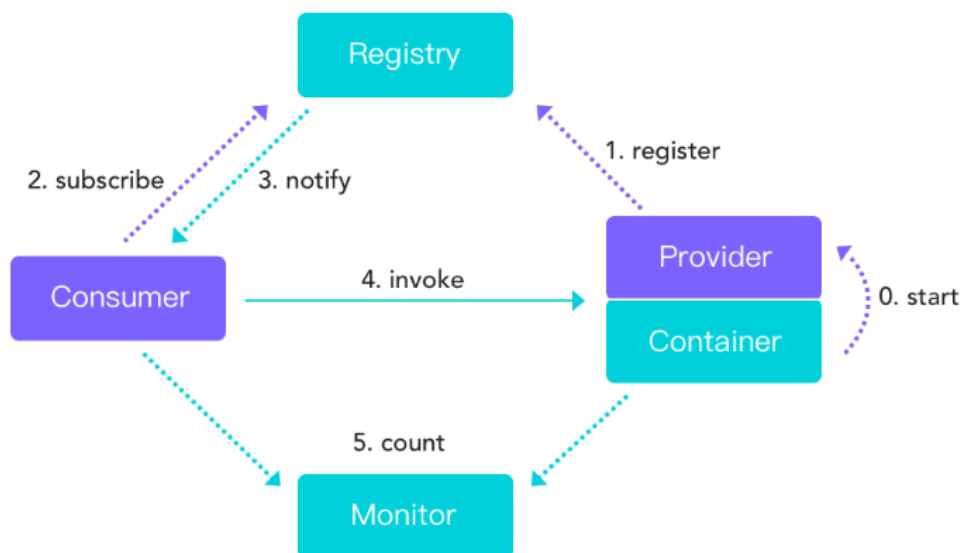
Dubbo是阿里巴巴在2011年开源的分布式服务框架，是服务化治理的核心框架。
Dubbo主要提供3个功能，远程接口调用，负载均衡和容错，服务注册和服务发现

- 2, Dubbo架构图

-

Dubbo Architecture

..... init async —> sync



- 3, 工作细节说明：

- 节点角色说明（掌握）

- 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 统计服务的调用次数和调用时间的监控中心
- Container 服务运行容器

- 调用关系说明

- 1. 服务容器负责启动，加载，运行服务提供者。
- 2. 服务提供者在启动时，向注册中心注册自己提供的服务。
- 3. 服务消费者在启动时，向注册中心订阅自己所需的服务。
- 4. 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。
- 5. 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。

- 6. 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。
- Dubbo 架构具有以下几个特点
 - 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
 - 健壮性
 - 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
 - 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, 搭建Zookeeper(动物管理员)注册中心
 - 安装JDK
 - 下载JDK的64位源码包，并将其上传到服务器上，或者采用wget的方式直接下载
wget url
 - 解压即可
 - 配置环境变量：vim /etc/profile
 -

```
# will prevent the need for merging in future

JAVA_HOME=/usr/local/soft/jdk1.8.0_181
PATH=$JAVA_HOME/bin:$PATH
export JAVA_HOME PATH
```

- 输入source /etc/profile，让环境配置生效
 - source命令通常用于重新执行刚修改的初始化文件，使之立即生效，而不必注销并重新登录。
- 安装Zookeeper
 - 上传
 - 解压
 - 修改zookeeper配置文件的名称：cp conf/zoo_sample.cfg conf/zoo.cfg
 - 编辑zookeeper主配置文件：vi conf/zoo.cfg（如果不配置zookeeper集群，则不需要做配置）
 - ```
The number of milliseconds of each tick
tickTime=2000
The number of ticks that the initial
synchronization phase can take
initLimit=10
The number of ticks that can pass between
sending a request and getting an acknowledgement
syncLimit=5
```
  - 启动zookeeper服务
    - ./bin/zkServer.sh start
    - ./bin/zkServer.sh stop
    - ./bin/zkServer.sh status
  - 开放防火墙端口
    - vim etc/sysconfig/iptables
    - service iptables restart（重启）
    - service iptables stop(强烈不推荐)
- 5，开发Dubbo应用
  - 1，Dubbo的特点
    - 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。
  - 2，项目的分布结构
    - 最外层创建一个maven工程，工程类型为pom，然后创建各个model
      - dubbo-interface
      - dubbo-provider
      - dubbo-comsumer
  - 3，引入依赖
    -

```

<!--引入dubbo的依赖-->
<dependency>
 <groupId>com.alibaba</groupId>
 <artifactId>dubbo</artifactId>
 <version>2.5.8</version>
</dependency>
<!-- zookeeper客户端依赖 -->
<dependency>
 <groupId>com.github.sgroschupf</groupId>
 <artifactId>zkclient</artifactId>
 <version>0.1</version>
</dependency>

```

- 4, dubbo-interface, 开发接口
- 5, dubbo-provider, 依赖dubbo-interface, 开发实现类, 并且配置spring文件, 发布服务

```

<?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: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">
 <!--1.定义应用的名称-->
 <dubbo:application name="java_dubbo_provider"> </dubbo:application>
 <!--2.定义服务注册中心-->
 <dubbo:registry protocol="zookeeper" address="192.168.10.168:2181">
</dubbo:registry>
 <!--3.用dubbo协议在20880端口暴露服务-->
 <dubbo:protocol port="20880"> </dubbo:protocol>
 <!--4.定义我们的实现类-->
 <bean id="userService" class="com.huangguizhao.service.UserServiceImpl"/>
 <!--5.声明暴露的服务接口-->
 <dubbo:service interface="com.huangguizhao.interfaces.IUserService"
ref="userService"/>
</beans>

```

- 通过启动Spring容器发布服务
  - System.in.read(); // 按任意键退出
- 6, dubbo-comsumer, 依赖dubbo-interface, 配置spring配置文件, 调用服务

- 配置spring

```

<?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: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">
 <!--1.定义应用的名称-->
 <dubbo:application name="java_dubbo_consumer"> </dubbo:application>
 <!--2.定义服务注册中心-->
 <dubbo:registry protocol="zookeeper" address="192.168.10.168:2181">

```

```
</dubbo:registry>
<!--3.生成远程服务代理，可以跟本地bean一样使用UserService-->
<dubbo:reference interface="com.huangguizhao.interfaces.IUserService"
id="userService"/>
</beans>
```

- 启动spring容器，调用服务
- 
- 6，安装监控中心
  - 上传，解压
  - 配置 vim conf/dubbo.properties，设置zookeeper注册中心的地址
  - 启动服务
    - ./bin/server.sh start
    - ./bin/server.sh stop
    - ./bin/server.sh restart
  - 查看日志
    - tail -f logs/stdout.log
  - 访问服务，查看具体信息
- 7，安装服务管理平台
  - 上传dubbo.war
  - 安装tomcat
    - 解压
    - 修改启动端口
    - 启动服务
  - 删除Root目录的内容
  - 将dubbo.war解压到Root目录下
    - unzip dubbo.war -d webapps/ROOT
  - 修改配置文件，配置zookeeper注册中心地址
    - vi webapps/ROOT/WEB-INF/dubbo.properties
  - 启动tomcat