

基于Dubbo的分布式系统架构视频教程



吴水成（水到渠成）

wu-sc@foxmail.com

840765167@qq.com

教程微信公众号：





基于Dubbo分布式系统架构

基础篇

使用Dubbo对传统项目工程进行服务化改造

--水到渠成

样例工程：传统的单工程项目（edu-demo）

Maven

J-UI

Struts2

Spring3

MyBatis3

Druid

MySQL5.6

只有一个用户信息表（增、删除、查、改）

改造成Dubbo服务调用方式后的工程结构

edu-common-parent (Maven父配置)

edu-facade-user (用户服务接口)

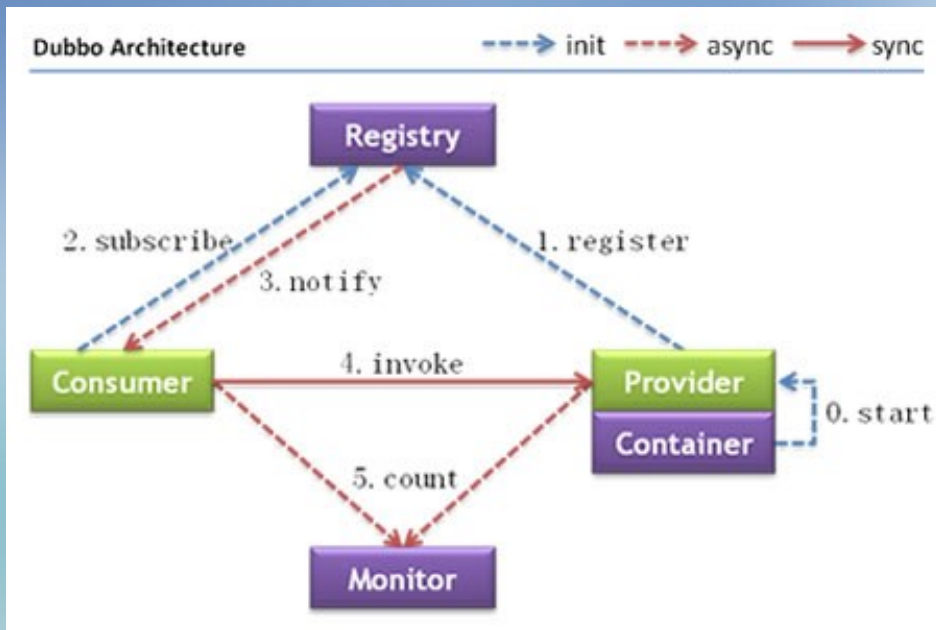
edu-service-user (用户服务实现)

edu-web-boss (服务消费者)

部署环境规划

192.168.2.61 edu-web-boss (consumer)
192.168.3.71 zookeeper-3.4.6 (注册中心)
192.168.3.72 edu-service-user (provider)
192.168.4.201 MySQL5.6 (数据库)

架构



•节点角色说明:

•**Provider:** 暴露服务的提供方

•**Consumer:** 调用远程服务的消费方

•**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:

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。

注册中心

可选方案：zookeeper、Redis

- 1、建议使用dubbo-2.3.3以上版本的使用zookeeper注册中心客户端
- 2、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。
- 3、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。

使用Dubbo进行规模服务化前的工程结构优化

一、使用Dubbo是为了实现系统的分布式服务化

二、做成分布式服务架构的项目特点：

- 1、多个服务
- 2、多种类型的工程
- 3、工程间需要相互调用
- 4、如何实现工程间解耦？（高内聚、低耦合）
- 5、工程该怎样拆分？
- 6、如何对大量的工程进行合理有效管理？（高级篇：持续集成）

使用Dubbo进行规模服务化前的工程结构优化

edu-common-parent (Maven父配置)

edu-common (公共工程)

edu-common-config (公共配置工程)

edu-common-core (公共core工程)

edu-common-web (公共web工程)

edu-facade-user (用户服务接口)

edu-service-user (用户服务实现)

edu-web-boss (服务消费者)