# 分布式基础知识

## 发展演变

|  |
| --- |
|  |

单一架构🡪垂直应用架构🡪分布式服务架构🡪流动式计算架构

## RPC

1. 定义：RPC是指远程过程调用，是一种进程间的通信方式

## dubbo简介

dubbo是一款高性能、轻量级的开源RPC框架，它提供了三大核心能力：面向接口的远程方法调用、智能容错和负载均衡、服务自动注册和发现

# Dubbo配置

## dubbo配置zookeeper注册中心

dubbo可以使用的注册中心有zookeeper、redis等，推荐使用zookeeper

zookeeper下载地址：<https://archive.apache.org/dist/zookeeper/zookeeper-3.4.13/>

注意：3.5版本的启动报错，这里使用3.4版本的

1. windows环境安装zookeeper，linux环境参考zookeeper使用文档
   1. 解压zookeeper安装包，修改配置文件zoo\_sample.cfg配置文件

重命名为zoo.cfg，并修改里面的dataDir路径

|  |
| --- |
|  |

其中zkData文件夹需要手动创建



|  |
| --- |
|  |

启动zookeeper，在zkServer.cmd对应路径的cmd命令窗口下输入zkServer.cmd回车

|  |
| --- |
|  |

说明服务端启动成功

1. 启动客户端测试zookeeper是否启动

|  |
| --- |
|  |

说明客户端启动成功

使用zookeeper命令进行测试

|  |
| --- |
|  |

说明可以正常使用

## Dubbo配置控制台

1. 下载项目并构建项目，下载地址：<https://github.com/apache/dubbo-admin>

|  |
| --- |
|  |

|  |
| --- |
|  |

解压并修改dubbo-admin项目下的application.properties里面zookeeper服务器地址，这里是我本地

|  |
| --- |
|  |

进入pom.xml目录命令下进行打包

|  |
| --- |
|  |

进入cmd命令窗口输入mvn clean package打包命令

|  |
| --- |
|  |

看到build success说明打包成功

|  |
| --- |
|  |

在相应的target目录下可以看到对应的jar包

|  |
| --- |
|  |

启动jar包

|  |
| --- |
|  |

端口是7001

|  |
| --- |
|  |

启动成功进行访问

|  |
| --- |
|  |

## Dubbo配置监控中心monitor:需要配置才能使用

（1）进入dubbo-monitor-simple的pom.xml目录命令下进行打包

|  |
| --- |
|  |

1. 在cmd命令窗口中执行打包命令

|  |
| --- |
|  |

1. 在target目录下可以看到生成的jar包

|  |
| --- |
|  |

1. 解压dubbo-monitor-simple-2.0.0-assembly.tar.gz并修改

dubbo-monitor-simple-2.0.0下面的conf下面的dubbo.properties配置文件

|  |
| --- |
|  |

主要是zookeeper地址及dubbo端口号

1. 启动dubbo的监控中心，双击assembly.bin下面的start.bat文件

|  |
| --- |
|  |

1. 进行访问localhost:8080

|  |
| --- |
|  |

## dubbo集成springboot

### 编写一个common模块，用于存放接口和实体类

1. 创建一个springBoot的maven工程，并导入依赖
2. 创建接口和实体类

### 编写一个生产者dubbo-user-service-provider

1. 创建一个springBoot的maven工程，并导入依赖

|  |
| --- |
| <!--引入dubbo的依赖--> <dependency>  <groupId>com.alibaba.boot</groupId>  <artifactId>dubbo-spring-boot-starter</artifactId>  <version>0.2.0</version> </dependency>  <!--引入common模块的依赖--> <dependency>  <groupId>com.liu.dubbo</groupId>  <artifactId>dubbo-service-common</artifactId>  <version>0.0.1-SNAPSHOT</version>  <scope>compile</scope> </dependency> |

1. 在application.yml中编写dubbo配置文件

|  |
| --- |
| dubbo:  application:  *# 指定当前服务/应用的名字* name: dubbo-user-service-provider  registry:  *# 指定注册中心的地址* address: 127.0.0.1:2181  *# 指定使用zookeeper作为注册中心* protocol: zookeeper  protocol:  *# 指定通信规则，协议和端口* name: dubbo  port: 20880  monitor:  *# 监控中心* protocol: registry |

1. 编写接口实现类，暴露服务(使用注解)

|  |
| --- |
|  |

1. 启动类里面添加注解@EnableDubbo开启基于注解的dubbo功能

|  |
| --- |
|  |

1. 启动生产者测试看是否注册到dubbo的控制台

|  |
| --- |
|  |

### 编写一个消费者dubbo-order-service-consumer

（1）创建一个springBoot的maven工程，并导入依赖

|  |
| --- |
| <!--引入dubbo的依赖--> <dependency>  <groupId>com.alibaba.boot</groupId>  <artifactId>dubbo-spring-boot-starter</artifactId>  <version>0.2.0</version> </dependency>  <!--引入common模块的依赖--> <dependency>  <groupId>com.liu.dubbo</groupId>  <artifactId>dubbo-service-common</artifactId>  <version>0.0.1-SNAPSHOT</version>  <scope>compile</scope> </dependency> |

（2）在application.yml中编写dubbo配置文件

|  |
| --- |
| dubbo:  application:  *# 指定当前服务/应用的名字* name: dubbo-order-service-comsumer  registry:  *# 指定注册中心的协议及地址* address: zookeeper://127.0.0.1:2181  monitor:  *# 监控中心* protocol: registry |

1. 编写接口实现类，以前我们用@Autowired，现在我们使用dubbo提供的注解@Reference注解来引用服务

|  |
| --- |
|  |

1. 启动类里面添加注解@EnableDubbo开启基于注解的dubbo功能

|  |
| --- |
|  |

1. 启动消费者测试看是否注册到dubbo的控制台

注意：在启动消费者之前一定要确保生产者启动成功

|  |
| --- |
|  |

1. 通过postman调用是否成功

|  |
| --- |
|  |

# 高可用

# Dubbo原理