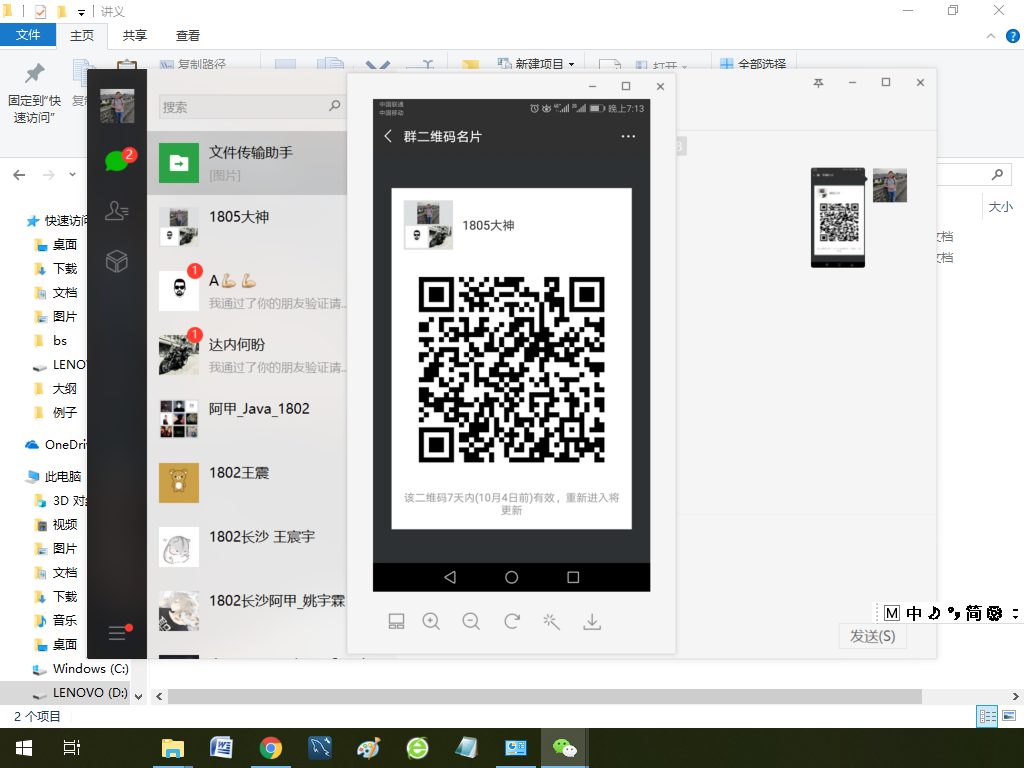
# 联系方式

张久军

13260051510



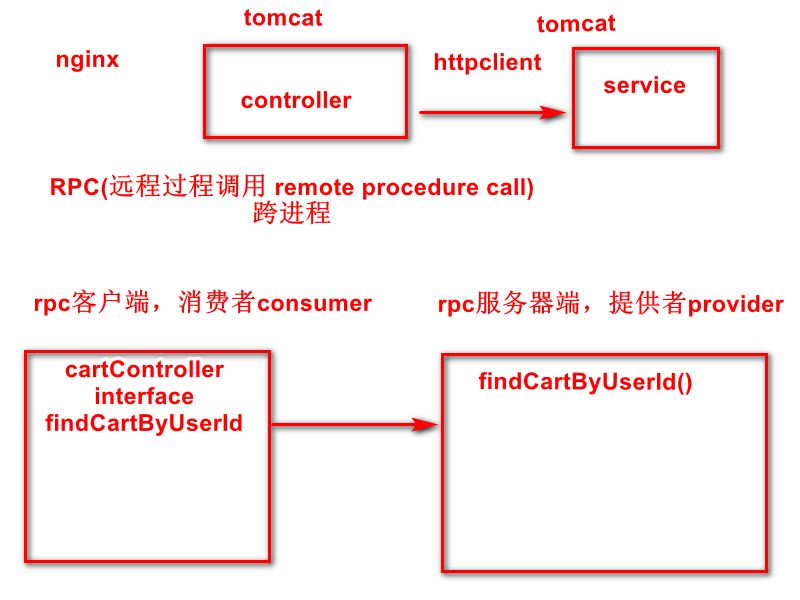
# 课程安排

1. 微服务dubbo,zookeeper
2. 消息队列rabbitmq
3. Solr
4. 容器docker
5. 手机端android

# rpc

Dubbo是微服务,服务是service

## 需求



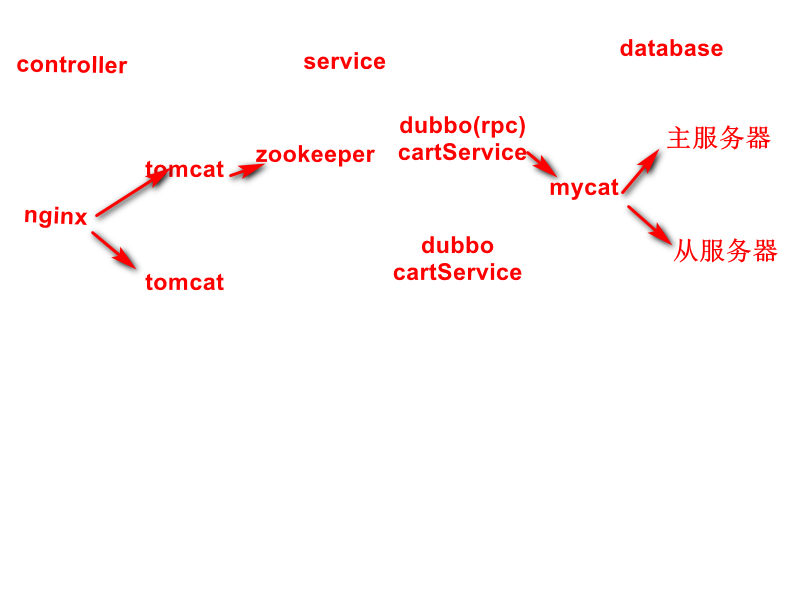
把controller和service放在一起的缺点

1. 不容易定位bug
2. 不容易扩展
3. 不能单独发布

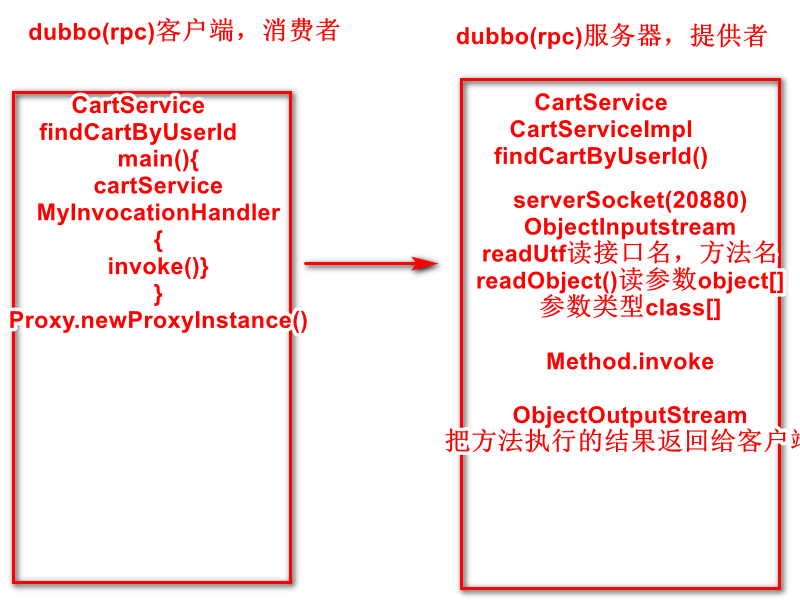
分布式优点

1. 容易定位bug
2. 容易扩展
3. 能单独发布

分布式实现



## 分析



## 设计

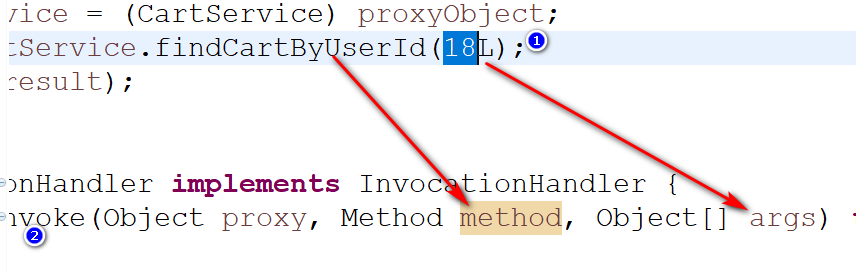
### Rpc server

1. Maven quick start
2. cartService
3. cartServiceImpl
4. providerMain

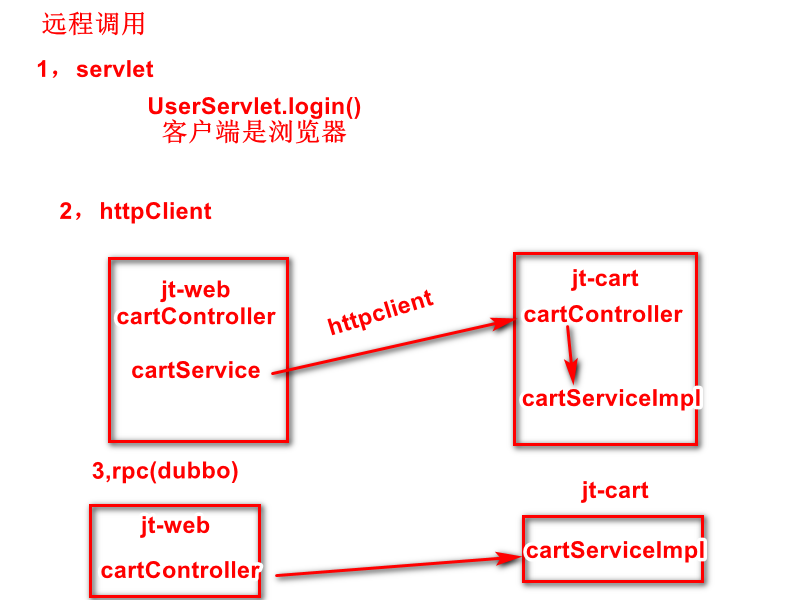
### Rpc client

1. maven quick start
2. cartService
3. CartController.invoke 测试
4. Invoke中联网 测试

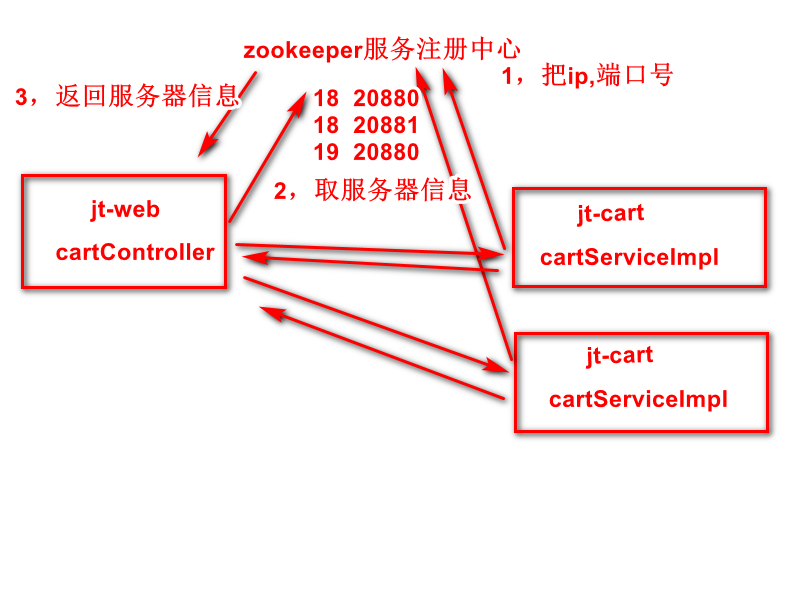
## 实现



## Rpc总结

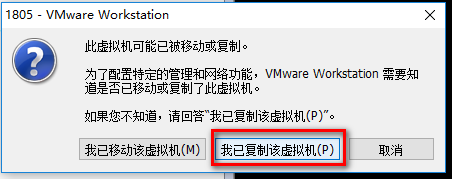


# Zookeeper



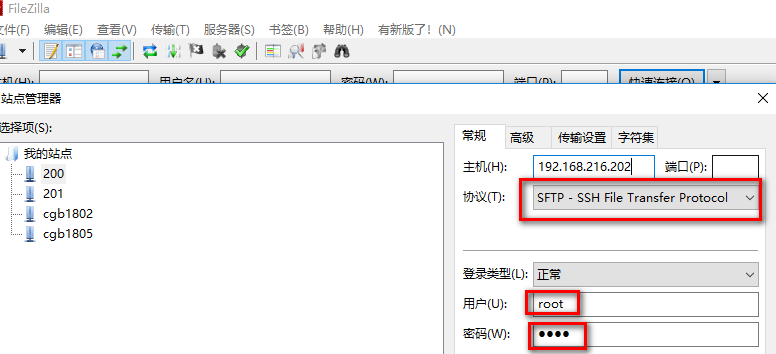
## 安装虚拟机

1. 解压 \培优课前资料\京淘课前资料下发-2\3-jt初始化虚拟机\ dsCentOS-empty.zip
2. 启动vmware
3. 双击解压后vmx文件，vmware就会加载虚拟机
4. 设置静态ip



5.xshell连接虚拟机

6.fz连接虚拟机



1. 在xshell执行命令关闭防火墙

service iptables stop

service iptables status

## 安装zookeeper

### 装java1.8

1. 使用fz 访问/usr/local/src
2. 创建文件夹java
3. 上传java1.8
4. 在xshell中解压java1.8

tar –xzvf

1. 使用fz访问/etc/profile修改java目录
2. 在xshell中执行source /etc/profile让环境变量生效
3. 在xshell中执行java -version

### 装zookeeper

1. 从ftp/source中下载zookeeper
2. 解压/usr/local/src/zookeeper.tar.gz
3. Xshell 中在/usr/local/src/zookeeper文件中执行 mkdir data log

cd zookeeper-3.4.8

[root@localhost zookeeper-3.4.8]# ls

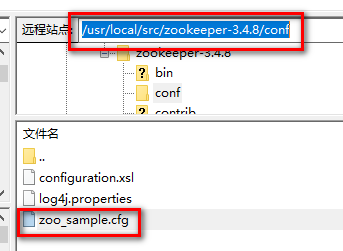
bin ivysettings.xml recipes

[root@localhost zookeeper-3.4.8]# pwd

/usr/local/src/zookeeper-3.4.8

[root@localhost zookeeper-3.4.8]# mkdir data log

1. 修改配置文件

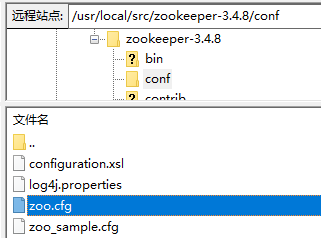


文件改成zoo.cfg 12行修改dataDir,增加dataLogDir

dataDir=/usr/local/src/zookeeper-3.4.8/data

dataLogDir=/usr/local/src/zookeeper-3.4.8/log

传到/usr/local/src/zookeeper-3.4.8/conf



1. 启动zookeeper服务器

[root@localhost zookeeper-3.4.8]# cd bin

[root@localhost bin]# ls

zkCli.sh zkEnv.sh zkServer.sh

[root@localhost bin]# sh zkServer.sh start

Starting zookeeper ... STARTED

1. 启动zookeeper客户端

Redis-server redis-cli

zkServer.sh zkCli.sh

进入zookeeper客户端

[root@localhost bin]# sh zkCli.sh

列出注册的服务

[zk: localhost:2181(CONNECTED) 0] ls /

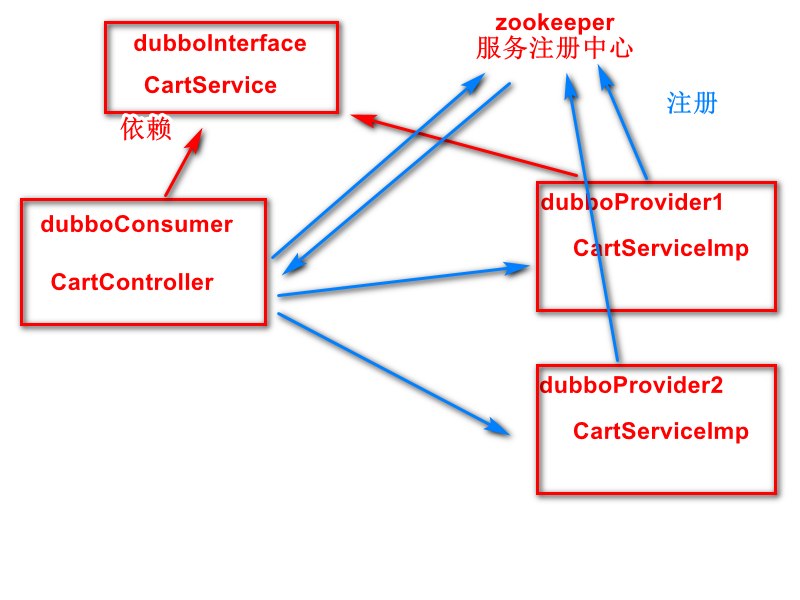
[zookeeper]

退出

[zk: localhost:2181(CONNECTED) 1] quit

# Dubbo案例

## 分析



## 设计

### dubboInterface

quickstart

创建interface CartService

### dubboProvider1

1. 创建maven
2. 依赖dubboInterface
3. 依赖dubbo,zookeeper
4. 实现类CartServiceImpl
5. applicationContext-provider.xml
6. 启动spring框架
7. 执行zkCli.sh

[zk: localhost:2181(CONNECTED) 0] ls /

[zookeeper]

[zk: localhost:2181(CONNECTED) 1] ls /

[dubbo, zookeeper]

查看 dubbo服务

[zk: localhost:2181(CONNECTED) 2] ls /dubbo

[com.tedu.service.CartService]

查看cartService有多少个提供者

[zk: localhost:2181(CONNECTED) 3] ls /dubbo/com.tedu.service.CartService/providers

### Dubbo后台管理，监控

不同jdk用不同的root.war

1. 查看windows 的jdk版本号
2. 把对应的root.war拷贝到tomat/webapps
3. 双击/bin/startup.bat启动tomcat,tomcat会把root.war解压到root中
4. 修改root/web-inf/dubbo.properties

dubbo.registry.address=zookeeper://192.168.216.202:2181

dubbo.admin.root.password=root

dubbo.admin.guest.password=guest

1. 重启tomcat
2. 浏览器访问http://localhost:8090

### dubboProvider2

1. 拷贝dubboprovider1项目
2. 修改pom.xml

<groupId>com.tedu</groupId>

<artifactId>dubboProvider2</artifactId>

1. 修改cartServiceImpl

**public** String findCartByUserId(Long userId) {

**return** "提供者2 返回"+userId;

}

1. 修改applicationContext-provider.xml

<!-- 1. 设置应用名称-->

<dubbo:application name=*"provider2-of-cart"*/>

<!-- 2.配置zookeeper地址 -->

<dubbo:registry address=*"zookeeper://192.168.216.202:2181"*>

</dubbo:registry>

<!-- 3.配置服务的端口号 -->

<dubbo:protocol port=*"20889"* name=*"dubbo"*>

</dubbo:protocol>

<!-- 4.配置实现类的类名 -->

<bean class=*"com.tedu.dubboProvider.CartServiceImpl"* id=*"cartService"*>

</bean>

<!-- 5.配置接口名,开放服务 -->

<dubbo:service interface=*"com.tedu.service.CartService"* ref=*"cartService"*>

</dubbo:service>

1. 启动服务器

**public** **static** **void** main(String[] args) {

//spring框架

ClassPathXmlApplicationContext context=**new**

ClassPathXmlApplicationContext("applicationContext-provider.xml");

System.***out***.println("start provider 2");

context.start();

**while**(**true**)

{}

}

1. 在dubbo后台管理中能看到提供者2

### dubboConsumer

1. quick start
2. 依赖dubboInterface
3. 依赖dubbo,zookeeper
4. 配置文件applicationContext-consumer.xml
5. 启动spring框架，调服务

## 实现

# 问题

跨进程和跨域

一台电脑上Tomcat访问mysql是跨进程

一台电脑上访问另一台电脑的进程

跨域

A网站用js(ajax)访问B网站的数据(json,xml)

A网站能访问B网站的js,image

<script src=baidu/jquery.js>

<img scr=baidu/1.png