

作业一作业二说明

1.运行环境

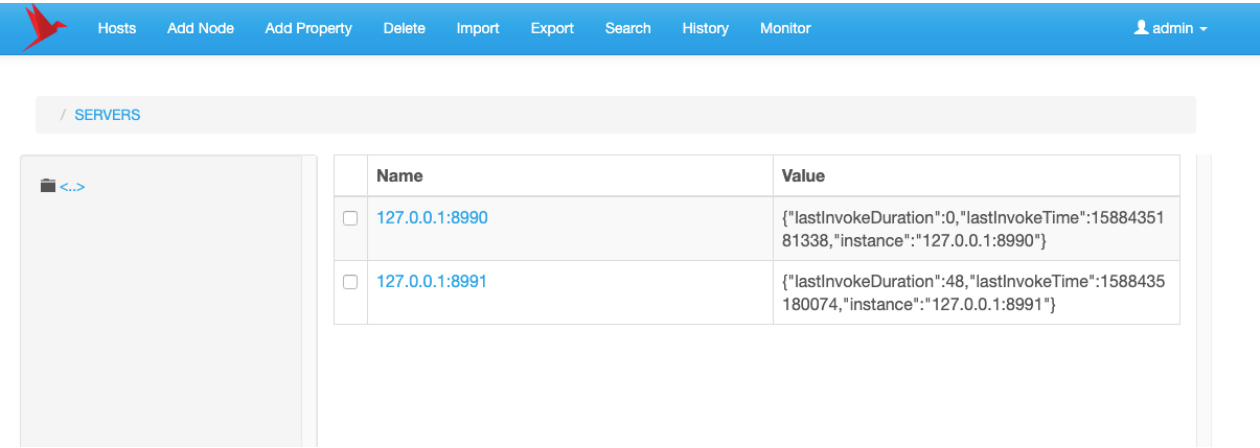
zookeeper： 3.6.0

2.需求

作业一

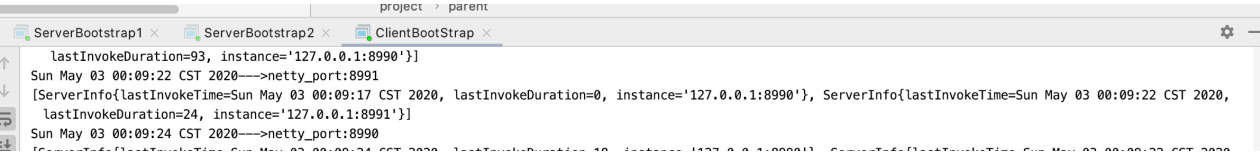
启动2个服务端，可以将IP及端口信息自动注册到Zookeeper

通过zkui工具查看zookeeper



客户端启动时，从Zookeeper中获取所有服务提供端节点信息，客户端与每一个服务端都建立连接

客户端获取server信息，并打印到控制台



某个服务端下线后，Zookeeper注册列表会自动剔除下线的服务端节点，客户端与下线的服务端断开连接

关闭8990服务端后，客户端只打印8991服务端信息，注册中心也只剩8991节点

IDE screenshot showing a Maven project structure with files like `pom.xml`, `zdy_rpc_provider.iml`, and `External Libraries`. The `dependencies` section of `pom.xml` is visible, showing a dependency on `log4j-core` with version `2.13.2`.

The console output shows logs for `ServerBootstrap1`, `ServerBootstrap2`, and `ClientBootStrap`. The logs indicate successful connections and data exchange between the server and client.

The bottom bar shows the application is running with the `admin` user.

/ SERVERS

Name	Value
<input type="checkbox"/> 127.0.0.1:8991	{"lastInvokeDuration":49,"lastInvokeTime":1588435963694,"instance":"127.0.0.1:8991"}

服务端重新上线，客户端能感知到，并且与重新上线的服务端重新建立连接

服务端上线后，自动注册到zookeeper中，客户端感知后拉取注册信息

IDE screenshot showing the `ServerBootstrap` class and its `NETTY_PORT` property. The console output shows logs for `ServerBootstrap1`, `ServerBootstrap2`, and `ClientBootStrap`. The logs indicate successful connections and data exchange between the server and client.

The bottom bar shows the application is running with the `admin` user.

/ SERVERS

Name	Value
<input type="checkbox"/> 127.0.0.1:8990	{"lastInvokeDuration":79,"lastInvokeTime":1588436234139,"instance":"127.0.0.1:8990"}
<input type="checkbox"/> 127.0.0.1:8991	{"lastInvokeDuration":46,"lastInvokeTime":1588436238305,"instance":"127.0.0.1:8991"}

作业二

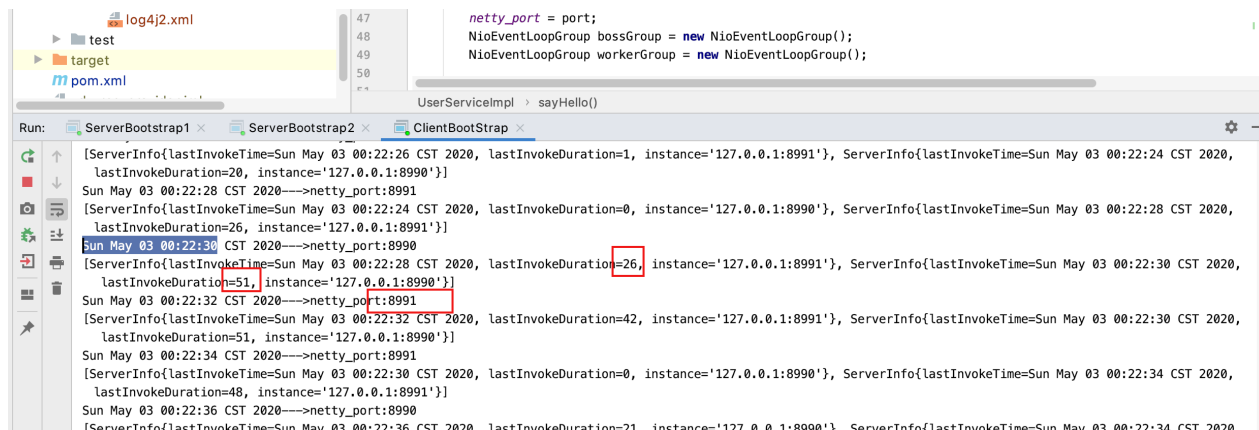
设置了一个随机模拟业务处理时长，客户端会根据时长选择上次处理时长短的服务端。处理时长有效期为5秒，超过5秒后，处理时长置为0

```
public String sayHello(String word) {
    System.out.println("调用成功--参数 "+word);
    int sleepTime = new Random().nextInt(100); //生成随机休眠时间，模拟业务处理时长

    try {
        Thread.sleep(sleepTime);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
    try {
        ServerInfo serverInfo = new ServerInfo(new Date(), sleepTime,
Constans.Zookeeper.INET_ADDRESS+": "+netty_port);

        com.lagou.ServerBootstrap.client.setData().forPath(Constans.Zookeeper.PREFIX
+"/" + Constans.Zookeeper.INET_ADDRESS+": "+netty_port,
            JSON.toJSONString(serverInfo).getBytes());
    } catch (Exception e) {
        e.printStackTrace();
    }
    return "netty_port:"+netty_port;
}
```

如图，8991上次处理时长为26，8990上次处理时长为51，故本次选择8991服务端



```
log4j2.xml
test
target
pom.xml

netty_port = port;
NioEventLoopGroup bossGroup = new NioEventLoopGroup();
NioEventLoopGroup workerGroup = new NioEventLoopGroup();

UserServiceImpI sayHello()

Run: ServerBootstrap1 x ServerBootstrap2 x ClientBootstrap x
[ServerInfo{lastInvokeTime=Sun May 03 00:22:26 CST 2020, lastInvokeDuration=1, instance='127.0.0.1:8991'}, ServerInfo{lastInvokeTime=Sun May 03 00:22:24 CST 2020, lastInvokeDuration=20, instance='127.0.0.1:8990'}]
Sun May 03 00:22:28 CST 2020-->netty_port:8991
[ServerInfo{lastInvokeTime=Sun May 03 00:22:24 CST 2020, lastInvokeDuration=0, instance='127.0.0.1:8990'}, ServerInfo{lastInvokeTime=Sun May 03 00:22:28 CST 2020, lastInvokeDuration=26, instance='127.0.0.1:8991'}]
Sun May 03 00:22:30 CST 2020-->netty_port:8990
[ServerInfo{lastInvokeTime=Sun May 03 00:22:28 CST 2020, lastInvokeDuration=26, instance='127.0.0.1:8991'}, ServerInfo{lastInvokeTime=Sun May 03 00:22:30 CST 2020, lastInvokeDuration=51, instance='127.0.0.1:8990'}]
Sun May 03 00:22:32 CST 2020-->netty_port:8991
[ServerInfo{lastInvokeTime=Sun May 03 00:22:32 CST 2020, lastInvokeDuration=42, instance='127.0.0.1:8991'}, ServerInfo{lastInvokeTime=Sun May 03 00:22:30 CST 2020, lastInvokeDuration=51, instance='127.0.0.1:8990'}]
Sun May 03 00:22:34 CST 2020-->netty_port:8991
[ServerInfo{lastInvokeTime=Sun May 03 00:22:30 CST 2020, lastInvokeDuration=0, instance='127.0.0.1:8990'}, ServerInfo{lastInvokeTime=Sun May 03 00:22:34 CST 2020, lastInvokeDuration=48, instance='127.0.0.1:8991'}]
Sun May 03 00:22:36 CST 2020-->netty_port:8990
[ServerInfo{lastInvokeTime=Sun May 03 00:22:36 CST 2020, lastInvokeDuration=21, instance='127.0.0.1:8990'}, ServerInfo{lastInvokeTime=Sun May 03 00:22:34 CST 2020,
```