



中国科学技术大学
University of Science and Technology of China



软件体系结构 实验报告

项目名称 基于 ZooKeeper+Dubbo 的分布式系统
年级专业 软件学院软件系统设计 2019 级
学生姓名 SA19225025 惠建诚 (队长)
学生姓名 SA19225027 李传国
学生姓名 SA19225028 李春国
学生姓名 SA19225029 李家毅

二零一九年 十二月 九日

目录

| | |
|----------------------|----|
| 一、 实验简介..... | 1 |
| 1. 实验目的..... | 1 |
| 2. 实验内容..... | 1 |
| 二、 实验需求..... | 1 |
| 1.功能性需求列表..... | 1 |
| 2.非功能性需求..... | 2 |
| 三、 相关技术介绍..... | 2 |
| 1. ZooKeeper 简介..... | 2 |
| 2. Dubbo 简介..... | 5 |
| 3. websocket 简介..... | 6 |
| 四、 项目分工..... | 7 |
| 五、 逻辑视图..... | 7 |
| 1. E-R 图..... | 7 |
| 2. 包图..... | 9 |
| 六、 过程视图..... | 10 |
| 1. 顺序图..... | 10 |
| 七、 开发视图..... | 12 |
| 八、 物理视图..... | 13 |
| 九、 场景..... | 14 |
| 十、 项目源码和运行截图..... | 15 |
| 1.项目源码..... | 15 |
| 2.运行截图..... | 15 |
| 十一、 项目总结..... | 19 |

图表目录

| | |
|---------------------------------|----|
| 表 2.1 功能性需求列表..... | 2 |
| 表 3.1 ZooKeeper 功能列表..... | 3 |
| 图 3.1 ZooKeeper 集群管理图..... | 4 |
| 图 3.2 ZooKeeper 节点数据操作流程图中..... | 4 |
| 图 3.3 Dubbo 架构流程图..... | 6 |
| 图 4.1 任务分工图..... | 7 |
| 图 5.1 E-R 图..... | 8 |
| 图 5.2 包图..... | 9 |
| 图 6.1 顺序图..... | 10 |
| 图 6.2 活动图..... | 11 |
| 图 7.1 配置视图..... | 12 |
| 图 8.1 部署图..... | 13 |
| 图 9.1 用例图..... | 14 |
| 图 10.1 zk 服务调用结果图..... | 15 |
| 图 10.2 输出 Excel 功能图..... | 16 |
| 图 10.3 输出的 Excel 表..... | 16 |
| 图 10.4 监测服务器功能图..... | 17 |
| 图 10.5 服务器结点数据打印图..... | 17 |
| 图 10.6 客户端 IP 信息展示图..... | 18 |
| 图 10.7 Dubbo 查看管理服务图..... | 18 |

一、实验简介

1.实验目的

以一个分布式门禁管理系统作为例子, 要求学生掌握使用各种结构视图, 实现对系统结构的描述, 并采用各种开源框架和具体编程技术实现该系统的简化版本。

2.实验内容

根据附录所列的用户需求列表, 整理用户需求, 完成用例图分析。根据附录提供的结构图, 采用课程所描述的各种视图方法, 完成对系统结构的描述。最后编程实现系统。

二、实验需求

1.功能性需求列表

| 序号 | 需求 | 内容 |
|----|---------|--------------------------------|
| I | II | III |
| 一 | 服务器功能模块 | 多用户的 B/S 架构, 数据库暂时采用 MySql |
| 1. | 控制命令传输 | 基于标准 Http 协议, 实现控制命令发送 |
| 2. | 机器状态接收 | 基于标准 Http 协议, 实现门禁状态信息的接收 |
| 3. | 管理配置 | 实现 IP 地址、同步信息的更新频率等各种设置信息的远程配置 |
| 4. | 数据查询 | 实现各种状态数据的查询汇总等 |
| 5. | 报表 | 实现以 EXCEL、PDF 等格式保存打印报表等 |

| | | |
|----|---------|---|
| 6. | 浏览器信息推送 | 有状态变化时向浏览器发送状态改变的信息 |
| 7. | 浏览器界面 | 采用 JavaScript 实现富客户端界面，如站点树、圆饼图、菜单等 |
| 8. | 心跳 | 接收基站控制器心跳数据信息，确保其是否在线 |
| 二 | 分布式机器 | |
| 1. | 控制命令 | 基于标准协议接收服务器发送的指令，实现基本控制 |
| 2. | 状态上传 | 基于标准协议实现各种机器状态的上传 |
| 3. | 远程配置 | 实现远端对 IP 地址、以及各种管理数据的配置 |
| 4. | 数据传输 | 因为本方案采用的是 WEB 服务器为中心的 B/S 架构，因此拟采用 Rest 和 Web Socke 等协议实现和服务器的数据传输，具有非常好的可扩展性和双向交互能力。 |
| 5. | 心跳 | 实现对服务器的心跳信息控制 |

表 2.1 功能性需求列表

2.非功能性需求

性能需求:响应时间迅速，数据传输快速。

可扩性需求:链接的站点能够扩展到几千个，能满足多用户并发访问。

灵活性需求：提供外部访问接口，能够实现后期如手机应用、外部应用的调用。程序升级简单。

三、相关技术介绍

1.ZooKeeper 简介

ZooKeeper 是一个分布式的，开放的分布式应用程序协调服务，是 Hadoop 和 Hbase 的重要组件。它是一个为分布式应用提供一致性服务的软件。在 ZooKeeper 中，Znode 是一个跟 Unix 文件系统路径相似的节点，可以往这个节点存储或获取数据。Zookeeper 使用 Watcher 察觉事件信息，当客户端接收到事件信息，比如连接超时、节点数据改变、子节点改变，可以调用相应的行为来处理数据。

ZooKeeper 主要优点包括：简单的分布式协调过程、同步 - 服务器进程之间的相互排斥和协作、有序消息、序列化（根据特定规则对数据进行编码，确保您的应用程序运行一致。）、可靠性、原子性：

ZooKeeper 主要功能如下：

| 功能 | 简介 |
|------------|---|
| 命名服务 | 按名称识别集群中的节点。它与 DNS 类似，但是用于节点。 |
| 配置管理 | 加入节点的系统配置信息的最新化。 |
| 群集管理 | 实时连接/离开群集中的节点和节点状态。 |
| 领导选举 | 选择节点作为协调目标的领导者。 |
| 锁定和同步服务 | 修改数据时锁定数据。此机制可帮助您在连接其他分布式应用程序（如 Apache HBase）时进行自动故障恢复。 |
| 高度可靠的数据注册表 | 数据即使在一个或几个节点关闭时的可用性 |

表 3.1 ZooKeeper 功能列表

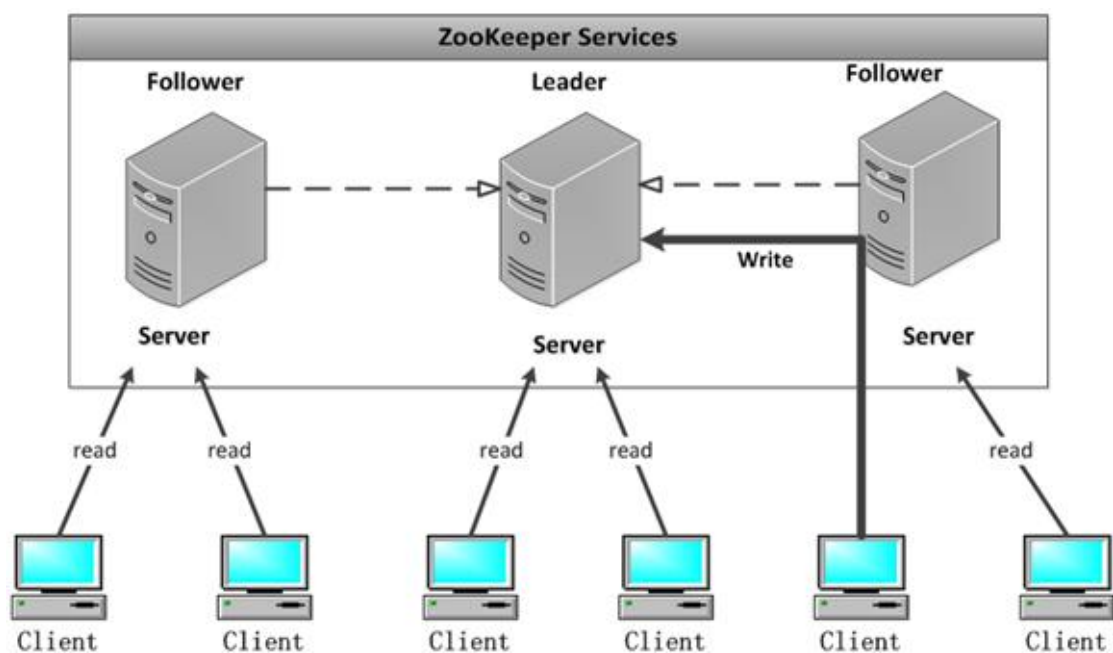


图 3.1 ZooKeeper 集群管理图

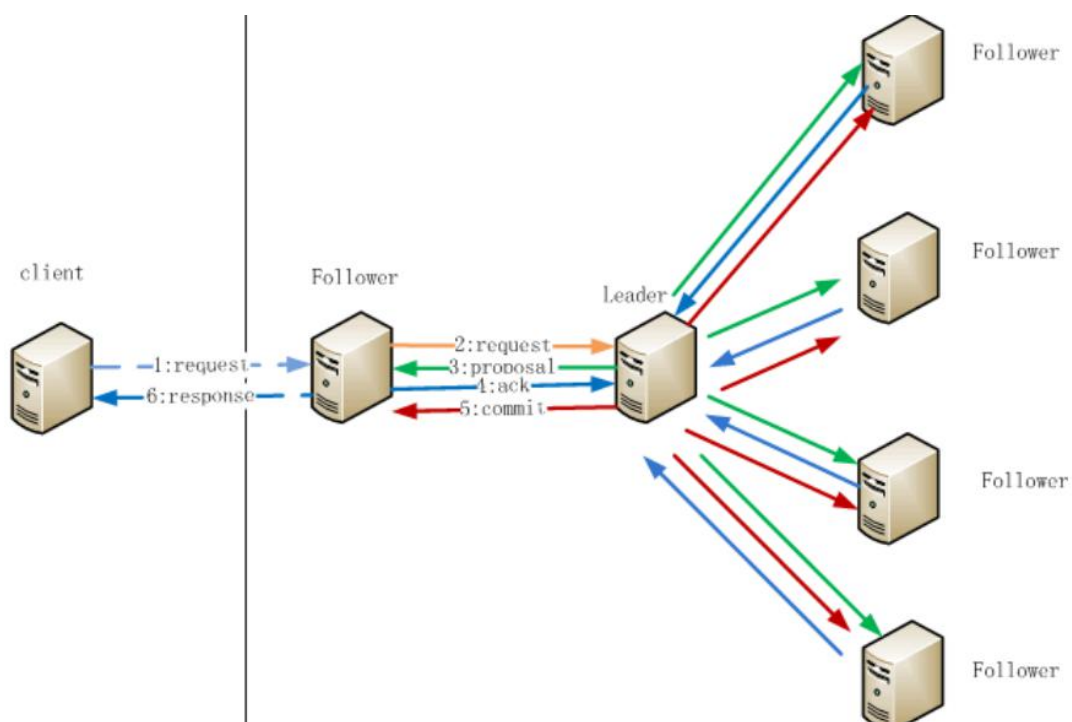


图 3.2 ZooKeeper 节点数据操作流程图

2.Dubbo 简介

Dubbo 是阿里开源的高性能优秀的服务框架，使得应用可通过高性能的 RPC 实现服务的输出和输入，可以和 spring 框架无缝集成。

Dubbo 是一款高性能、轻量级的开源 Java 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，以及服务自动注册和发现。

Dubbo 架构的优点包括：

- a. 透明化的远程方法调用。像调用本地方法一样调用远程方法；只需简单配置，没有任何 API 侵入。
- b. 软负载均衡及容错机制。可在内网替代 nginx lvs 等硬件负载均衡器。
- c. 服务注册中心自动注册 & 配置管理。不需要写死服务提供者地址，注册中心基于接口名自动查询提供者 ip。使用类似 ZooKeeper 等分布式协调服务作为服务注册中心，可以将绝大部分项目配置移入 ZooKeeper 集群。
- d. 服务接口监控与治理。Dubbo-admin 与 Dubbo-monitor 提供了完善的服务接口管理与监控功能，针对不同应用的不同接口，可以进行 多版本，多协议，多注册中心管理。

Dubbo 架构流程图如下：

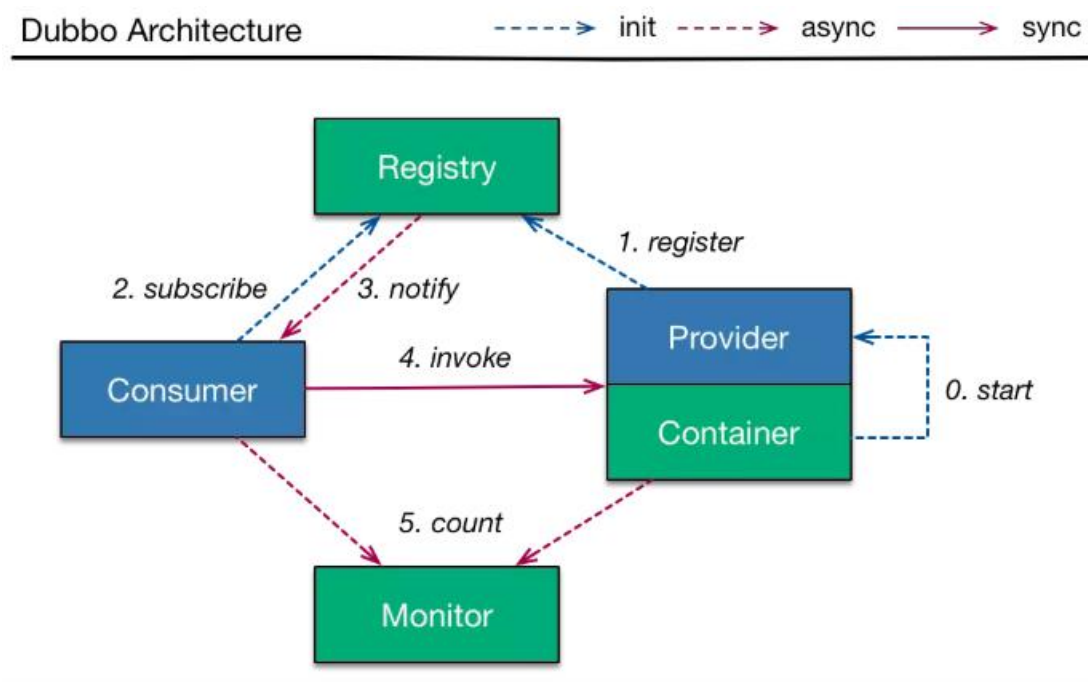


图 3.3 Dubbo 架构流程图

Provider: 服务提供方，本次项目中为 SA-service 项目。

Consumer: 服务消费者，本次项目中为 SA-controller 项目

Registry: 注册中心，本次项目中为 ZooKeeper。

Monitor: 统计服务调用次数和调用时间的监控中心。

3.websocket 简介

我们一直使用的 Http 协议只能由客户端发起，服务端无法直接进行推送，这就导致了如果服务端有持续的变化客户端想要获知就比较麻烦。WebSocket 协议就是为了解决这个问题应运而生。

WebSocket 是一种在单个 TCP 连接上进行全双工通信的协议。WebSocket 使得客户端和服务端之间的数据交换变得更加简单，允许服务端主动向客户端推送数据。在 WebSocket API 中，浏览

器和服务器只需要完成一次握手，两者之间就直接可以创建持久性的连接，并进行双向数据传输。

四、项目分工

本次项目主要包括项目实现和文档编写两部分，具体分工如下：

| 人员 | 工作 |
|---------|---|
| 惠建诚(组长) | 项目的整体设计、项目框架的搭建、项目的整体编码、项目配置以及项目测试。 参与实验报告的编写， 配置文档的编写。 |
| 李传国 | 学习相关知识。 项目中打印 Excel 表格功能的实现。参与实验报告的编写。 |
| 李春国 | 学习相关知识。 参与实验报告的编写。 |
| 李家毅 | 学习相关知识。 参与实验报告的编写。 |

图 4.1 任务分工图

五、逻辑视图

1.E-R 图

本分布式系统逻辑视图主要描述系统应该向用户提供的服务。系统被分解成一组密钥抽象，（大部分）以对象或类的形式表

示。本系统的类和对象采取抽象、封装和继承的原则。

本系统首先采取 E-R 图描述本系统在用例图所描述的相应功能。

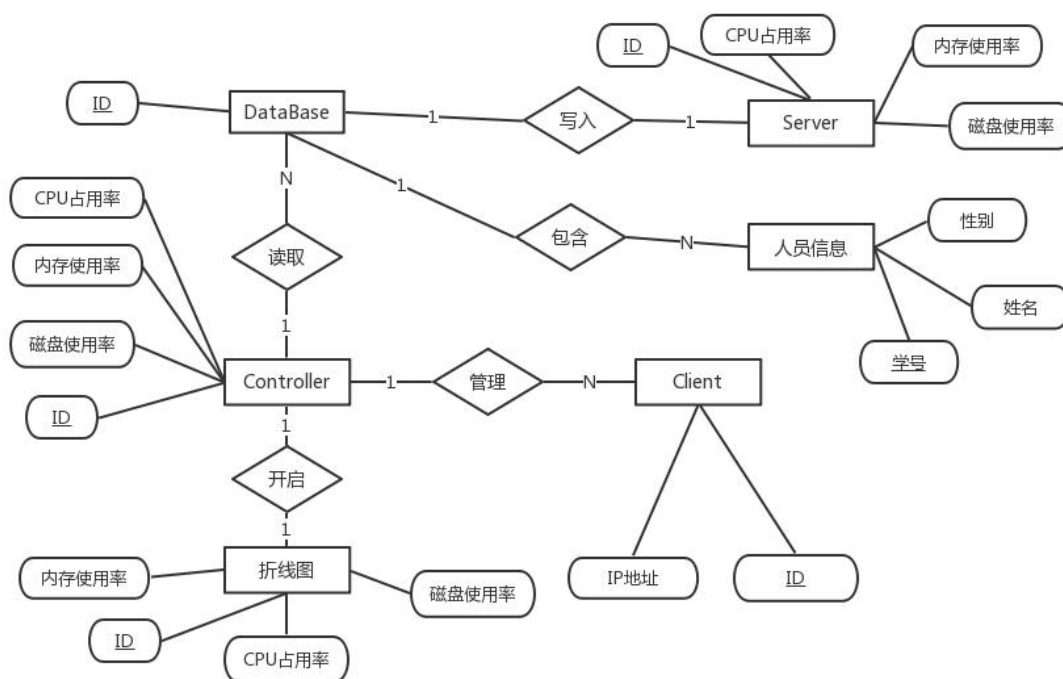


图 5.1 E-R 图

由图看出本系统公有 6 个实体。其中人员信息实体主要介绍了客户的性别、姓名、学号等个人基本信息。而 DataBase 实体只是包含数据库的 ID 属性，它主要作用为存储人员信息，本系统只有一个数据库，一个数据库可以存储多个客户信息。Controller 实体包含自身电脑的 CPU 占用率、内存使用率、磁盘使用率和本身的 ID 它可以读取 DataBase 中的人员信息，并且可以开启折线图。折线图主要包括 Controller 中所查询到的信息，并将这些信息动态展现出来。Client 实体主要存放本机的 IP 地址和 ID 号，并且其信息可以被 Controller 调用，侦测到。Sever 实体含自身电脑的 CPU 占用率、内存使用率、磁盘使用率和本身的 ID。并且可以将已知信

息写入 DataBase 中。

2.包图

本系统使用 UML 方法借助包关系图和类模板来表示逻辑体系结构。本系统的包图显示一组包及其逻辑关系：关联、用法、组合、继承等。

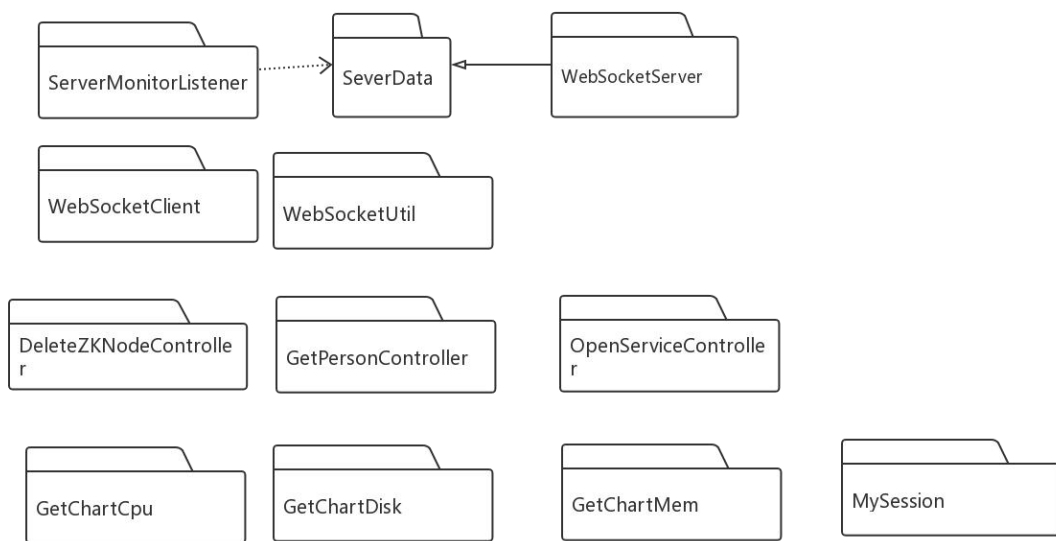


图 5.2 包图

本系统中的包大多设置的可以自给自足因此并不存在太多关联、用法、组合、继承等逻辑关系。主要就是 WebSocketServer 类继承了 SeverData 类，而 ServerMonitorListener 类实现了 SeverData 类。

六、过程视图

1.顺序图

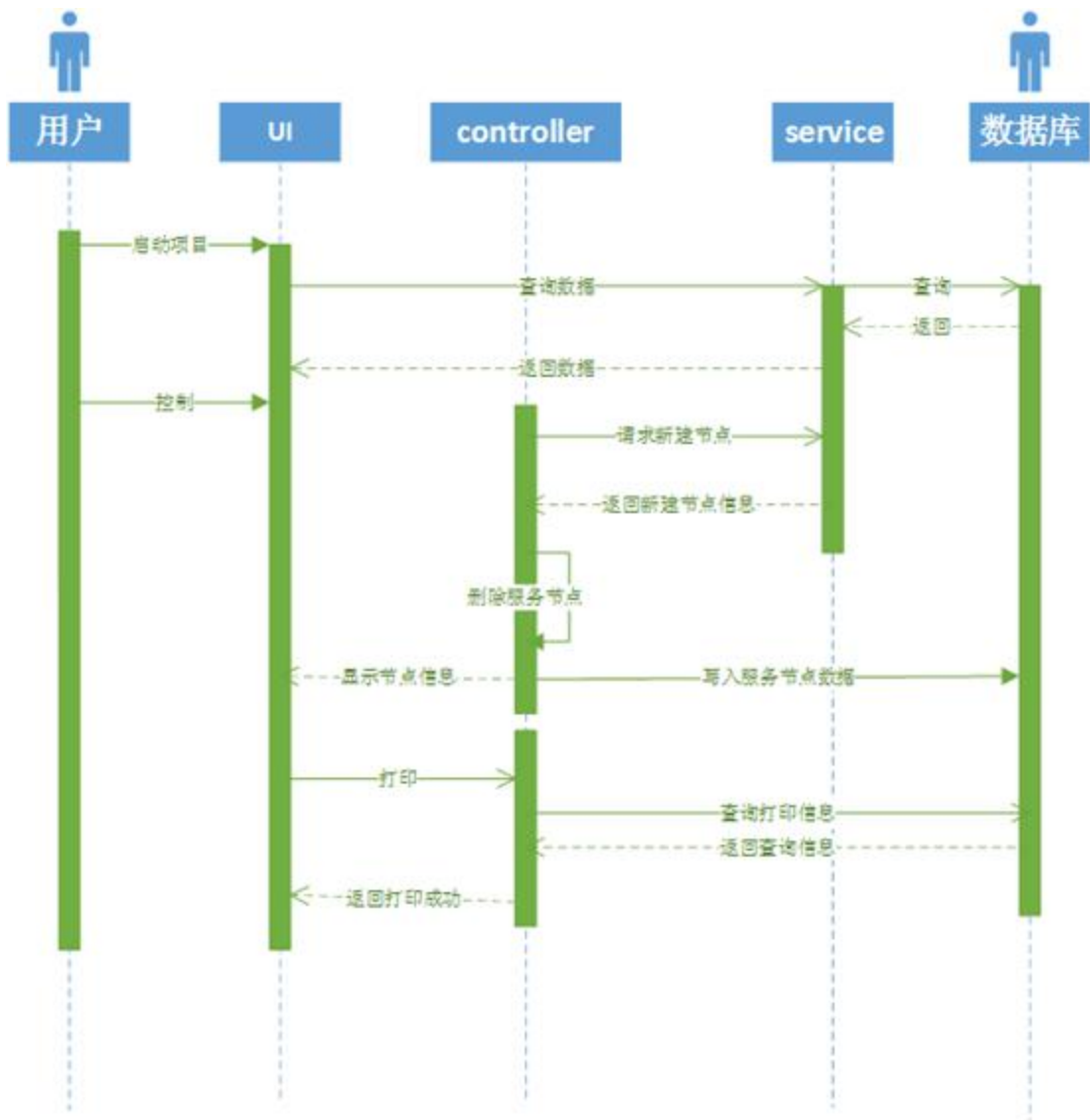


图 6.1 顺序图

如上顺序图所示，当用户启动项目之后，系统进入 UI 界面，用户通过控制 UI 界面来完成相关操作。用户可以通过向 service 发送查询请求，service 收到请求后向客户端数据库查询相关信息并将信息传回到 UI 界面上。同时在 zookeeper 启动后，controller 可以向 service 请求新建服务节点，之后 service 在 zookeeper 上新

建节点并将节点信息返回给 controller 并通过表格和折线图的方式显示在 UI 界面上。controller 也可以删除已经存在的服务节点，同时，controller 会将服务节点的信息写入到已连接的客户端数据库中。用户也可以通过 UI 界面向 controller 发送打印命令，可以将数据库中的相关信息以 excel 的形式导出到本地中。

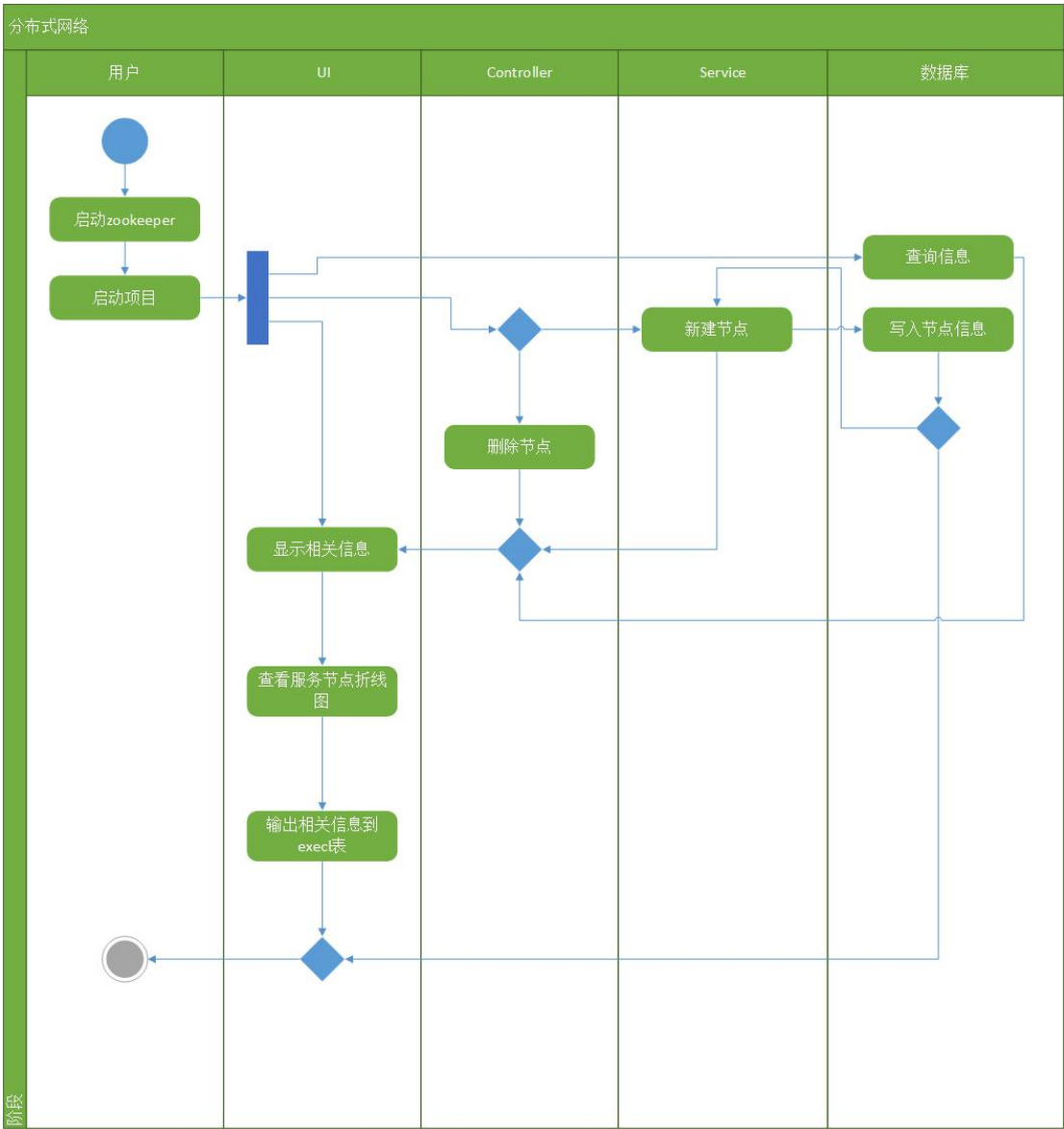


图 6.2 活动图

七、开发视图

配置视图关注的是软件开发环境下实际模块的组织 (受开发难度、软件管理、重用性和通用性及由工具集、编程语言所带来的限制)。在此我们可以用 UML 的包图来进行描述。

在本次分布式项目中，我们创建了 SA-common 包用于声明接口，Controller 包中存放的是监听客户端请求的代码，UI 包可调用 Controller 包从而实现多个客户端的实时监听和查看所有客户端 CPU 的使用情况。Service 包用于实现接口，并与 Zookeeper 交互实现服务的注册，同时 Service 通过前后端的 API 接口从数据端取数据，并以折线图的显示呈现给用户，如下是本次分布式的配置视图。

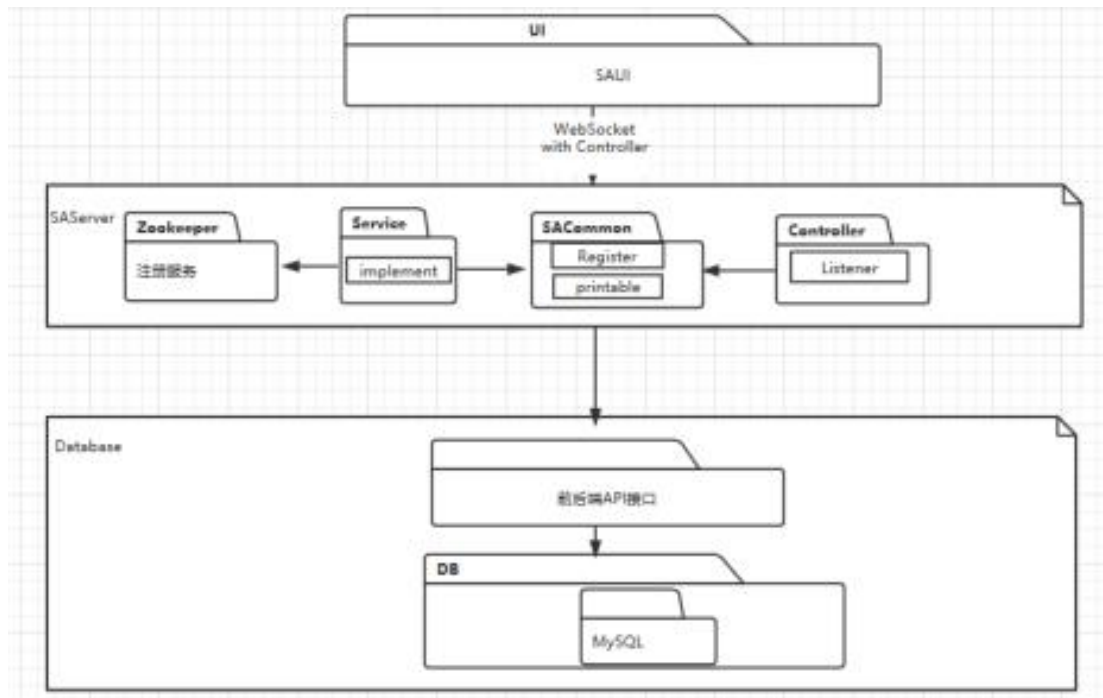


图 7.1 配置视图

八、物理视图

物理视图对应自身的结构实现建模，类会映射成物理结构中的结点，关注的是依赖于硬件的非功能性需求，如可用性、可靠性、性能吞吐量等，我们可以用 UML 的部署图来进行描述。

在本次分布式项目中，实现的构件主要是获取人员信息，增加服务结点和 Zookeeper 注册服务。UI 端通过 Websocket 调用服务器端服务，服务器端可以实时监测客户端请求，同时也可以从数据库端读取数据，将数据反馈给客户端，并在 UI 界面上显示出来，如下是本次分布式的物理视图。

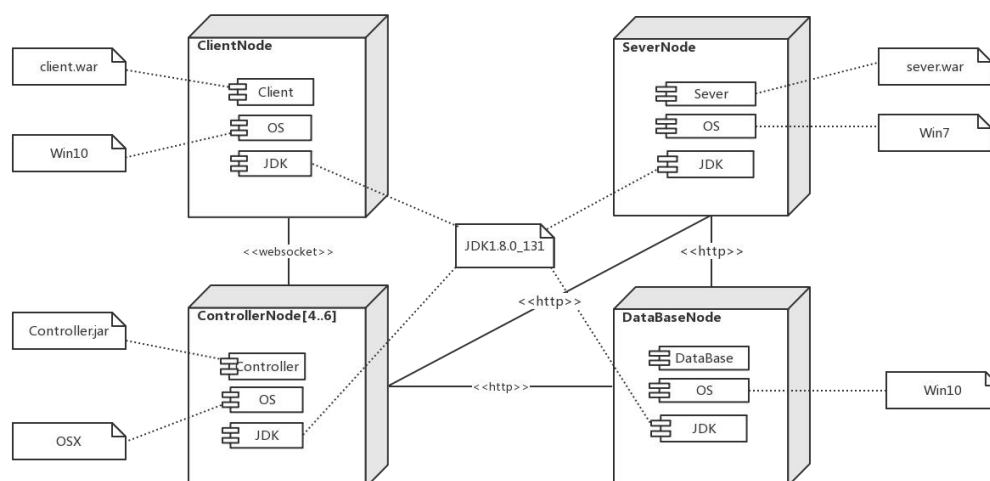


图 8.1 部署图

由图可以看出，本系统共分为 4 种节点，分别是服务器节点、客户端节点、数据库节点、控制器节点。服务器与客户端基于 WebSocket，控制器和数据库间的命令传输、服务器和数据库间的命令传均基于标准 Http 协议实现。并且本系统所采用的 JDK 版均相同均为 1.8.0_131 版。

九、场景

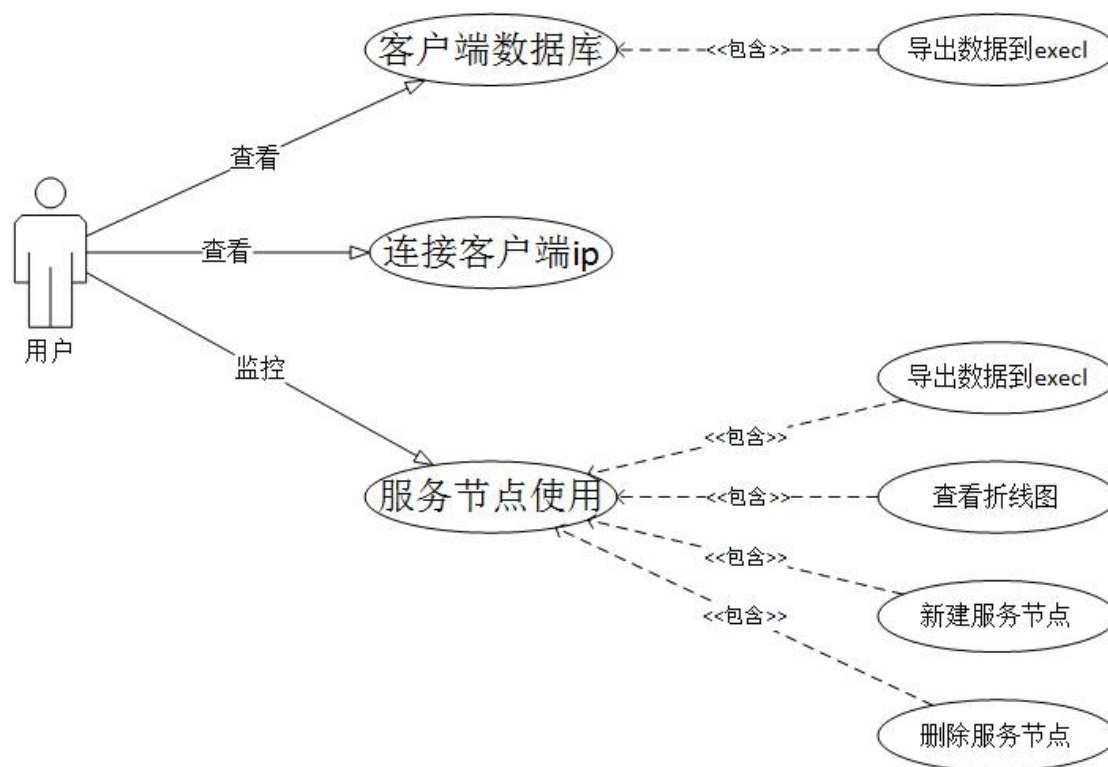


图 9.1 用例图

如上面所示用例图，在 zookeeper 服务器开启后，当客户端连接上服务器之后，用户便可以访问客户端的数据库，并且可以将数据库中的数据以 excel 表的形式导出到电脑本地中。同时客户还可以查看所有连接到服务器中的客户端的 IP 地址。并且可以创建服务节点供给客户端使用，并可以通过折线图和表格显示服务点的使用情况，包括 ID，CPU，MEM，DISK。并且可以将其写到数据库中以及以 excel 表的形式导出到电脑本地中。使用过程中客户也可以新建和删除服务节点。

十、项目源码和运行截图

1.项目源码

源码地址: <https://github.com/ustc-sse-hjc/SA-parent.git>

2.运行截图

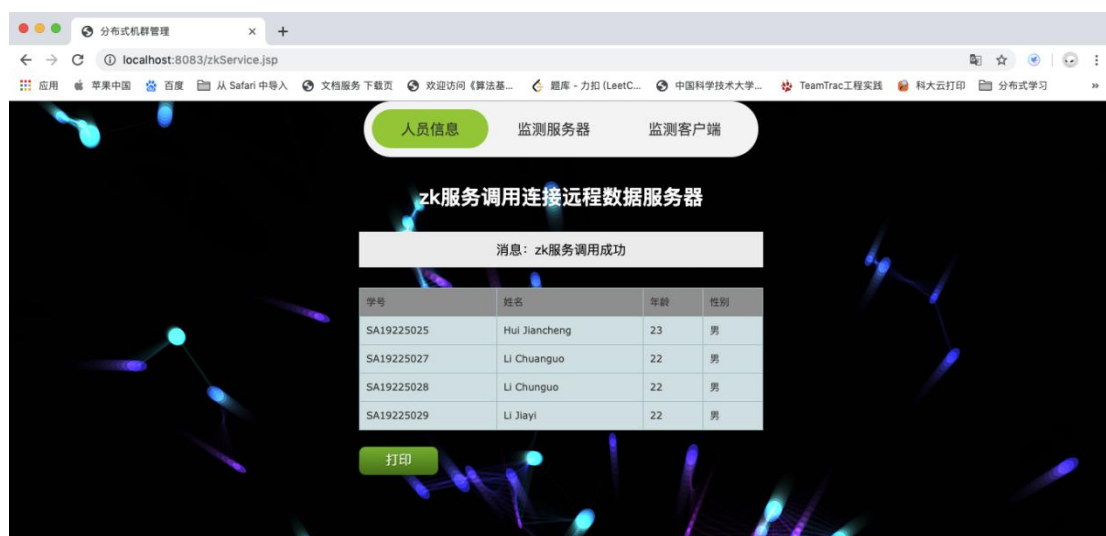


图 10.1 zk 服务调用结果图

图 10.1 是对 ZooKeeper 中注册的人员信息查询服务的调用结果图。

其中核心代码为:

```
ApplicationContext context = new FileSystemXmlApplicationContext(  
    new String[] { "dubbo-consumer.xml" });  
  
final PersonService personService =  
(PersonService)context.getBean("personService");
```

扫描 dubbo-consumer.xml 文件, 通过 id 调用注册的服务。



图 10.2 输出 Excel 功能图

| | A | B | C | D | E | F |
|---|------------|---------------|----|----|---|---|
| 1 | ID | 姓名 | 年龄 | 性别 | | |
| 2 | SA19225025 | Hui Jiancheng | 23 | 1 | | |
| 3 | SA19225027 | Li Chuanguo | 22 | 1 | | |
| 4 | SA19225028 | Li Chunguo | 22 | 1 | | |
| 5 | SA19225029 | Li Jiayi | 22 | 1 | | |
| 6 | | | | | | |
| 7 | | | | | | |

图 10.3 输出的 Excel 表

图 10.2 展示的是打印功能，图 10.3 展示的是打印功能的结果。

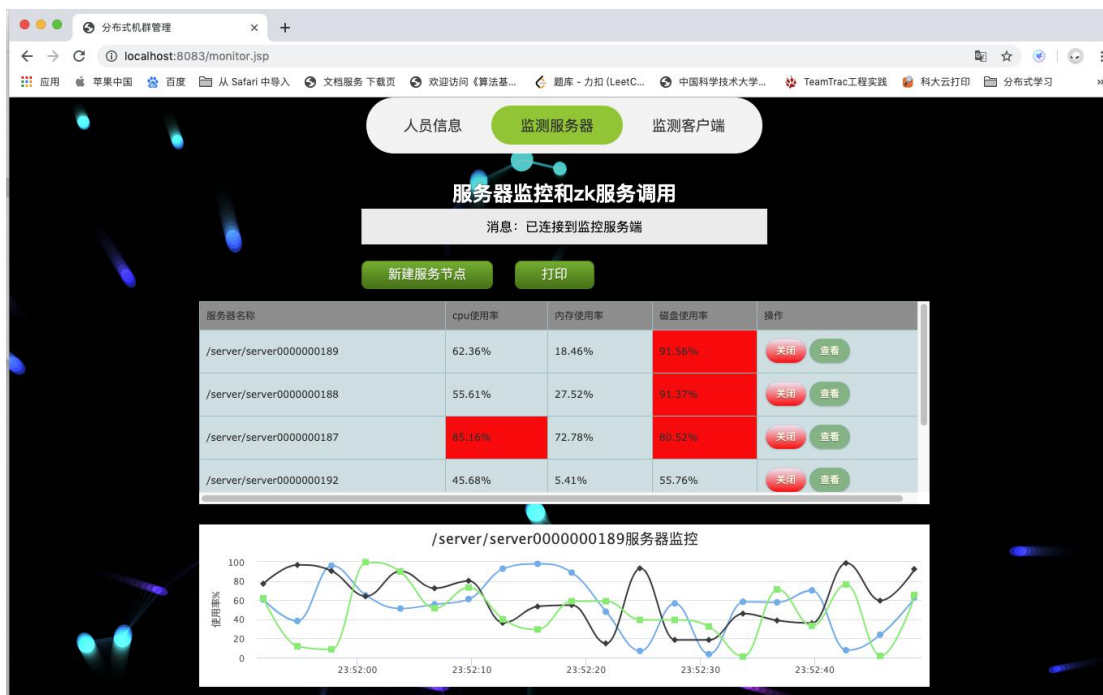


图 10.4 监测服务器功能图

图 10.4 展示的是本次项目的核心功能，对服务节点的信息监测，高于阈值的数据红色高亮显示，并且信息是 1 秒钟更新一次的，实时展示到界面上。点击查看可以显示数据的可视化曲线图。

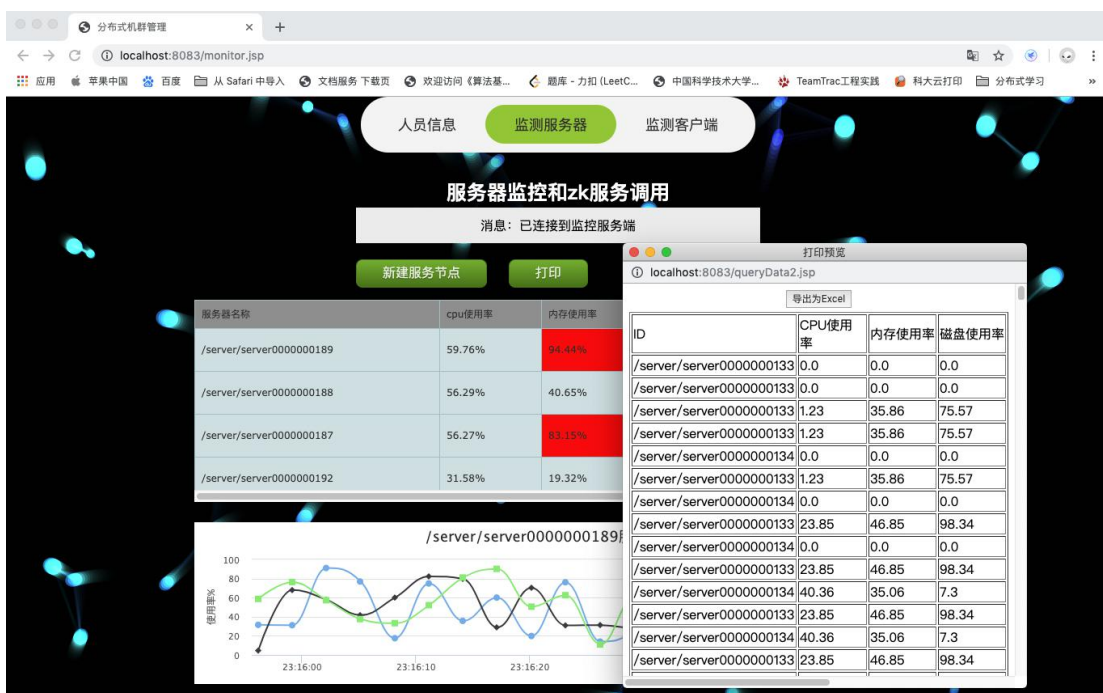


图 10.5 服务器结点数据打印图

图 10.5 展示的是服务器结点 cpu、内存、磁盘使用率的历史数据打印功能。



图 10.6 客户端 IP 信息展示图

图 10.6 展示的是对客户端 IP 地址监测的功能。在界面上可以实时显示客户端的 IP 地址，并且实时更新。

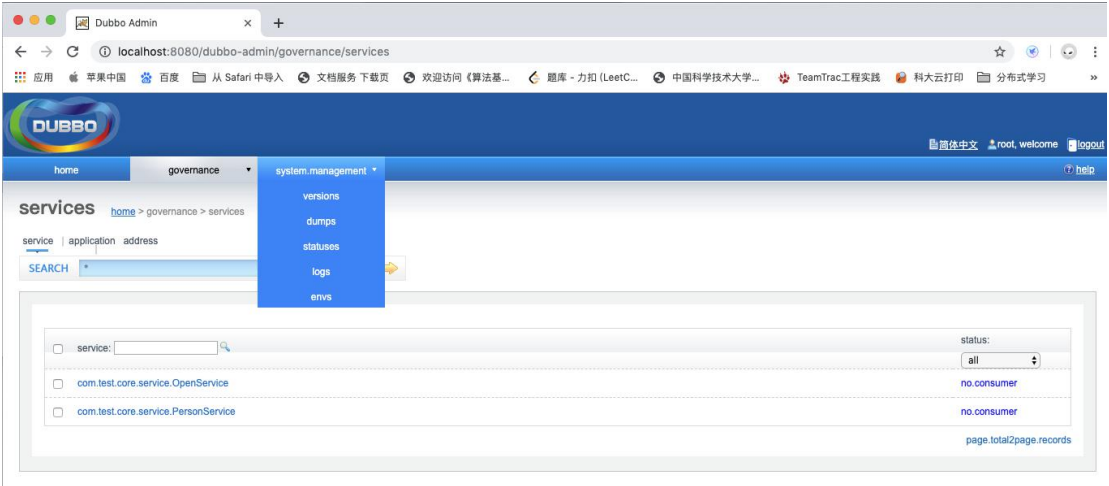


图 10.7 Dubbo 查看管理服务图

图 10.7 展示的是 Dubbo 提供的一个用于查看和管理 Dubbo 服务的 web 服务 Dubbo-admin。在该页面可以显示注册的服务的信息和调用服务的信息。

十一、项目总结

惠建诚总结

这次项目的实现过程让我学习到了很多知识,也让我初步了解了分布式系统领域。拿到实验要求真的是一头雾水,因为以前没有了解过分布式系统,更别说写个项目出来。所以在和学长无数次邮件讨论以及查阅资料后,最终定下来用 ZooKeeper+Dubbo+WebSocket 这一个架构来实现。

ZooKeeper 是一个开放的分布式应用程序协调服务,为分布式应用提供一致性服务的软件。它的主要功能包括配置管理,名字服务,提供分布式同步以及集群管理。每个功能的简介都在相关技术介绍模块中提到了,在这就不赘述了,本次项目主要是利用了集群管理的功能,当一个服务节点出现变动时,它会触发 ZooKeeper 的 Watcher 事件监听机制,并且会把相应的数据传到客户端,这就满足了实验要求的监听功能,同时,节点删除时,同样也会触发 Watcher 事件,把信息传导客户端,所以这也实现了实验中要求的心跳协议同样的功能。对于 ZooKeeper,我还学到了 ZooKeeper 集群中 Leader 的选举以及节点数目最好为奇数的原理等等。我认为这次项目的实现也只是使我对 ZooKeeper 刚刚入门,对于 ZooKeeper 更深知识的学习还需要我继续努力。

Dubbo 在上文也提到了是阿里开源的高性能优秀的服务框架,使得应用可通过高性能的 RPC 实现服务的输出和输入,可以和 spring 框架无缝集成。本次项目采用的框架正是 SSM 框架,所以可以和 Dubbo 进行很好的集成。Dubbo 通过配置文件,向注册中心注册服务,客户端只要从注册中心调用服务即可。本次的注册中心即为 ZooKeeper,在 SA-service 项目中,注册的服务有两个,在 SA-controller 里通过每个 service 特定的 id 即可进行相关服务的调用。同时 Dubbo 提供一个用于查看和管理 Dubbo 服务的 web 服务 Dubbo-admin。这个项目的配置因为各种原因花费了我很长时间,最终配置成功后,我们

可以在这个 web 项目中看到 Dubbo 服务的各种信息, 在项目截图中可以看到。

WebSocket 就不用多说了, 这次主要是用在 SA-UI 项目中, 实现客户端和服务端都可以主动的推送消息, 同时也可实现对数据的实时刷新, 结合 websocket 传来的数据和 HighChart, 绘制了这次项目中实时变化的曲线图。不过各种连接超时以及连接冲突的问题也是很让人头疼, 将每个 onOpen、onMessage、onClose 设置成异步的也是会一些问题, 所以 WebSocket 功能很强大, 还有很多知识点需要我去继续学习。

总结一下, 这次项目让我初步学习到了 ZooKeeper、Dubbo 和 WebSocket, 真的是受益匪浅。同时它也让我学会了使用 HighChart 绘制曲线图 (我觉得这个实时变化的曲线图是我这次项目中最让我满意的功能)。分布式系统还有太多太多需要我去学习的知识, 我还需要继续努力学习。

李传国总结

“我们是一个 team” 这句话在本次实验项目中我深有体会。本次实验项目是要求完成一个分布式集群系统, 这是一个转专业过来的我从来没有接触过的, 本次实验能顺利完成离不开小组里每个人的努力。通过本次实验我也学会了很多这方面的相关知识。

首先, 是关于数据库方面的知识。原本通过暑假的学习, 我以为我已经会使用数据库了, 但是直到实验开始我才明白我啥也不会。光会直接建表是没用的, 我还需要知道如何在项目开发中去调用它。在本次实验中, 使用的开发语言是 java 语言。通过查询知道可以使用 jdbc 来调用数据库中的数据, 好不容易能够访问自己电脑上的数据库了, 但是实验要求的是访问其余电脑上的数据库, 通过自己的努力终于解决了。还有就是学会了如何将读到 jsp 页面上的数据库数据导出到本地中。通过编写相关 script 方法, 最后成功将读到的数据导出到本地电脑中。

其次是学会了分布式的相关知识。本次实验中, 我们组使用的是

ssm 框架, ssm 框架是 Spring + Spring MVC + MyBatis 的缩写, 是继 SSH 之后, 目前比较主流的 Java EE 企业级框架, 适用于搭建各种大型的企业级应用系统。Spring 就像是整个项目中装配 bean 的大工厂, 在配置文件中可以指定使用特定的参数去调用实体类的构造方法来实例化对象。Spring 的核心思想是 IoC (控制反转), 即不再需要程序员去显式地`new`一个对象, 而是让 Spring 框架帮你来完成这一切。SpringMVC 在项目中拦截用户请求, 它的核心 Servlet 即 DispatcherServlet 承担中介或是前台这样的职责, 将用户请求通过 HandlerMapping 去匹配 Controller, Controller 就是具体对应请求所执行的操作。SpringMVC 相当于 SSH 框架中 struts。mybatis 是对 jdbc 的封装, 它让数据库底层操作变的透明。mybatis 的操作都是围绕一个 sqlSessionSessionFactory 实例展开的。mybatis 通过配置文件关联到各实体类的 Mapper 文件, Mapper 文件中配置了每个类对数据库所需进行的 sql 语句映射。在每次与数据库交互时, 通过 sqlSessionSessionFactory 拿到一个 sqlSession, 再执行 sql 命令。

同时我也了解到 Zookeeper 的相关知识, 它是本次项目开发中使用的一个重要的软件。ZooKeeper 是一个分布式的, 开放源码的分布式应用程序协调服务, 是 Google 的 Chubby 一个开源的实现, 是 Hadoop 和 Hbase 的重要组件。它是一个为分布式应用提供一致性服务的软件, 提供的功能包括:配置维护、域名服务、分布式同步、组服务等。

通过本次实验项目的开发, 我学会了很多知识也增加了一些项目开发的经验。同时我在这里需要感谢我的组员, 能够成功完成本次实验项目, 离不开每个组员的努力。

李春国总结

在本次分布试验中学到了很多专业性的知识。通过本次分布式系统制作首先让我系统的了解了 zookeeper 基本功能和特性。在本次实验中我了解了 zookeeper 的集群自身的特性一个 zookeeper 服务器集群, 一开始就会进行选主, 主节点挂掉后也会进行选主。使用 zab 协议中的选主机制进行选主, 也就是每个节点进行一次提议, 刚开始提

议自己，如果有新的提议则覆盖自己原来的提议，不断重复，直到有节点获得过半的投票。完成一轮选主。并且在本次实验我的电脑一直会出现会出现错误，每次调试时候总会报错，关于 tomcat，但是最开始检测时候又显示 tomcat 已经安装好，因此并没有往 tomcat 的环境变量配置去考虑，在每次我的电脑出现错误时候，我的队员们他们都很热心帮助我，帮我想办法，热心的指导我，让我感受到了很深的温暖，并且最后也是在我的队员的帮助下发现的环境变量的问题。然后经过配置环境变量，并且把 jdk 的安装路径和 tomcat 的安装路径复制到 startup.bat 里面之后解决了此问题。在本次本不是实验中，我也学到了关于页面设计、制作和布置的一些基本知识。学习会了 html、css 并且对 js 也有了一定的了解。通过这个分布式试验也使我对 div 块元素有了更深刻的理解，很多时候网页的制作都是通过若干个块元素进行堆叠的，即使一个标签并不是块元素也可以通过一些处理使其变成块元素或者行内成块。这样将更方便的对其进行处理。经过这段时间的想学习也使我能够通过 html+css 制作一些简单的小网页。虽然这个小网页目前还无法实现轮播图片的效果，但是对于我来说已经很不错了。

我们四人于偶然中相知，科大中相遇，南区中相识，525 中相知。525 夜晚的曙光圈圈点点着我们四人的记忆，兜兜转转，是你们替我播散眼前的迷雾，给我一片开阔视野，使我迷茫的心充满坚定，落地生根，即将产生硕果。感谢有你们。WAF! 525 🐮🍺。

李家毅总结

通过本次分布式项目，我大概了解到了分布式项目搭建的具体流程，我主要学到了 SQL 语言的基本语法和 Zookeeper 实现注册服务的相关知识。

ZooKeeper 是一个分布式的，开放源码的分布式应用程序协调服务，它是一个为分布式应用提供一致性服务的软件，提供的功能包括：配置维护、域名服务、分布式同步、组服务等。ZooKeeper 的目标就是封装好复杂易出错的关键服务，将简单易用的接口和性能高效、功

能稳定的系统提供给用户。在本次项目中我们实现了向 ZooKeeper 注册服务，通过 SAcCommon 包向 Zookeeper 中取服务，通过 WebSocket 反馈给用户，从而实现分布式。

MySQL 是一种关系数据库管理系统，关系数据库将数据保存在不同的表中，MySQL 所使用的 SQL 语言是用于访问数据库的最常用标准化语言。本次主要是利用 SQL 基本语法，向本地数据库中建立人员信息表，从而客户端可以从数据库中取出相关表格，并实现打印。

另外，最重要的是在本次项目中从这个 Team 中我感受了团队做项目时的那种激情，这是以前所没有的，这将激发团队中的每个人的积极性，我们是最棒的!!!