哈尔滨工业大学

**计算学部**

**2024年秋季学期**

**《软件架构与中间件》课程**

**实验报告**

**Lab 1：简易消息中间件开发**

|  |  |  |
| --- | --- | --- |
| **姓名** | **学号** | **联系方式** |
| 余昊卿 | 2023120253 | Felixupri@gmail.com |
| 马嘉良 | 2023120259 | 1009728013@qq.com |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **学号：** | **2023120253** | **姓名：余昊卿** |  | |
| **学号：** | **2023120259** | **姓名：马嘉良** |  | |
| 实验概述 | | | |  |
| 实验目的 1）学习事件系统型软件架构风格  2）学会使用观测者模式或者订阅发布模式对事件系统进行建模分析  3）能够独立开发简单的消息中间件  4）能够灵活应用所开发的中间件到实际系统 实验要求 1）2人结对成组  2）不可借助开源消息中间件框架  3）模拟实现一种简单的消息中间件，能达到对系统功能解耦的目的  4）分析消息中间件的吞吐率等非功能指标  5）应结合《软件过程与工具》课程中进销存系统(或其他实际软件系统)进行分析，给出哪些场景可以运用该消息中间件  6）应该出设计过程和实现细节 | | | | |
| 设计方案 | | | |  |
| 中间件采用订阅发布模式，类图如下：    基于socket编程实现了服务器和客户端之间的通信。采用单例模式，Broker和同名的消息队列只会创建一个实例。订阅者可以订阅若干消息队列，发布者发布新消息时，中间件会通知订阅者，时序图如下： | | | | |
| 程序实现 | | | |  |
| 开发环境与工具 | | | | |
| 开发语言：Python 3.10  开发环境：Windows 11 | | | | |
| 实现要点说明 | | | | |
| Broker采用单例模式实现，程序运行过程中仅创建一个Broker。    Broker的主要接口实现如下：    Subscriber启动一个线程用于接收消息： | | | | |
| 中间件测试与应用 | | | |  |
| 中间件测试与验证 | | | | |
| 运行Broker，并创建队列Q1和Q2。    创建Subscriber A，订阅Q1的消息。    创建Subscriber B，订阅Q2的消息。    订阅成功后运行Broker的终端会显示相应的信息：    向Q1发布消息：    运行Broker的终端和运行Subscriber的终端会显示相应的消息，Subscriber A接收到了来自Q1的消息。      向Q2发布消息：    Subscriber B接收到了来自Q2的消息。      **测试消息中间件的吞吐量**  可以通过测试发布1000条消息的耗时来衡量消息中间件的吞吐量，测试结果如下： | | | | |
| 实际系统中的运用 | | | | |
| **电商网站的订单处理**  当用户在电商平台上下单时，系统需要执行一系列操作，例如生成订单、处理支付、更新库存、发送订单确认邮件等。这些操作如果同步执行，会导致用户下单时延迟过长。通过中间件订单生成操作可以立即返回，而支付处理、库存更新等操作可以异步地推送到消息队列中，后台任务逐步处理这些任务。使用中间件降低了用户操作的延迟，提升用户体验，同时可以通过水平扩展消费者来处理更多订单。  **图片或视频的批量处理**  对于在线图片分享网站，用户上传图片后，需要对图片进行压缩、打水印等处理。这些处理任务较为耗时，且不适合在上传请求时同步完成。通过使用中间件，图片上传服务可以立即返回，然后异步处理这些任务。使用中间件可以让任务处理更具弹性，可以通过增加或减少消费者来调节并发量和资源使用，保证任务处理的平稳性。  **新闻推送系统**  在新闻门户网站中，当有新的新闻发布时，用户可能订阅了不同类别的新闻推送。通过中间件的发布/订阅模型，可以方便地将新闻发布到不同的消费者。发布/订阅模型使得生产者和消费者解耦，生产者不需要知道哪些用户订阅了特定类型的新闻，中间件负责消息的路由和分发。 | | | | |
| 结对开发过程记录 | | | |  |
| **（1）角色切换与任务分工** | | | | |
| 表1-1结对开发角色与任务分工 | | | | |
| | 日期 | 时间 | 驾驶员 | 领航员 | 本段时间的任务 | | --- | --- | --- | --- | --- | | 9/25 | 16:20-17:10 | 余昊卿 | 马嘉良 | 讨论中间件的应用场景与实现功能，初步完成整体架构设计。 | | 9/26 | 15:26-18:37 | 马嘉良 | 余昊卿 | 实现中间件的核心功能 | | 9/27 | 13:00-16:30 | 马嘉良 | 余昊卿 | 进一步完善中间件 | | 10/3 | 18:20-23:56 | 余昊卿 | 马嘉良 | 测试中间件，撰写实验报告。 | | | | | |
| **（2）工作日志** | | | | |
| 由领航员负责记录，记录结对开发期间的遇到的问题、两人如何通过交流合作解决每个问题的。 | | | | |
| 表1-2 结对开发工作日志 | | | | |
| | 日期/时间 | 问题描述 | 最终解决方法 | 交流过程 | | --- | --- | --- | --- | | 9/25 | 中间件的接口设计 | 参考RabbitMQ进行设计 | 分别阅读了常见中间件如Kafka等的文档，后经过交流确定方案。 | | 9/27 | 如何实现服务器和客户端之间的通信 | 基于Socket编程实现 | 考虑到基于Socket编程有较高的灵活性，于是确定该方案。 | | | | | |
| **（3）结对开发工作现场照片** | | | | |
| 请其他同学帮助拍摄结对开发现场照片至少2张或讨论记录截屏。 | | | | |
| 结对开发现场照片1 结对开发现场照片2 | | | | |
| 实验总结 | | | |  |
| 通过本次实验，进一步了解了事件系统型软件架构风格及其应用场景，学习了观测者模式和订阅发布模式的原理，并对事件系统进行建模与分析。通过事件驱动的方式可以实现系统的高效解耦，提高系统的灵活性。通过开发一个简单的消息中间件，实现中间件的基本功能，增强了对事件驱动架构的理解。目前简易中间件的性能有待提高，系统的健壮性仍有不足。我们会在未来的学习与实践过程中继续学习相关知识，逐步完善该系统。 | | | | |
| 教师评语 | | | |  |
|  | | | | |