

后端开发的一些问题探讨

浙江大学城市学院

彭 彬

pengb@zucc.edu.cn

本课程后端开发的学习目标

1) 巩固Java程序开发技术

包括了基本的程序设计的理解； “恰当地”使用Java的库、常见的设计方法、Java开发工程环境

2) 掌握JavaEE基础~Servlet

包括了Servlet的构造、理解Servlet容器、配合Http协议的工作模式

3) 掌握IoC思想


4) 掌握SpringMVC程序开发

RESTful Api + MyBatis + MySQL、理解MVC模式、理解Spring JPA

5) 后端架构及扩展（可选）：消息队列、MongoDB、ElasticSearch

6) 完成一个完整的基于SpringMVC的应用程序

动手实现一个具有具体功能的应用程序；



前后端分离

前后端是什么？

应用于某个用户终端设备的部分是前端，提供业务计算/数据处理服务的视为后端。常见的前端可以分为两类：基于浏览器或者基于特定终端操作系统。特定终端操作系统典型的是比如Android、iOS或者一些物联网设备的操作系统，当然还有火热的车机系统。（很多基于各种Linux系统，比如树莓派的系统）

分离的是什么？

分离的是前后端的交互方法，即如何设计前端程序与后端程序的交互（数据交换的方式）。这个概念特定与基于浏览器的前端设计。基于特定操作系统开发的前端天生就是前后端分离的。我们课程讨论的分离是通过Http协议 + Json数据格式的模式，区别于过去后端直接生成Html的做法。这个模式反过来影响了前后端程序的结构设计（后端不在有UI代码、前端使用复杂的JavaScript构建应用程序）

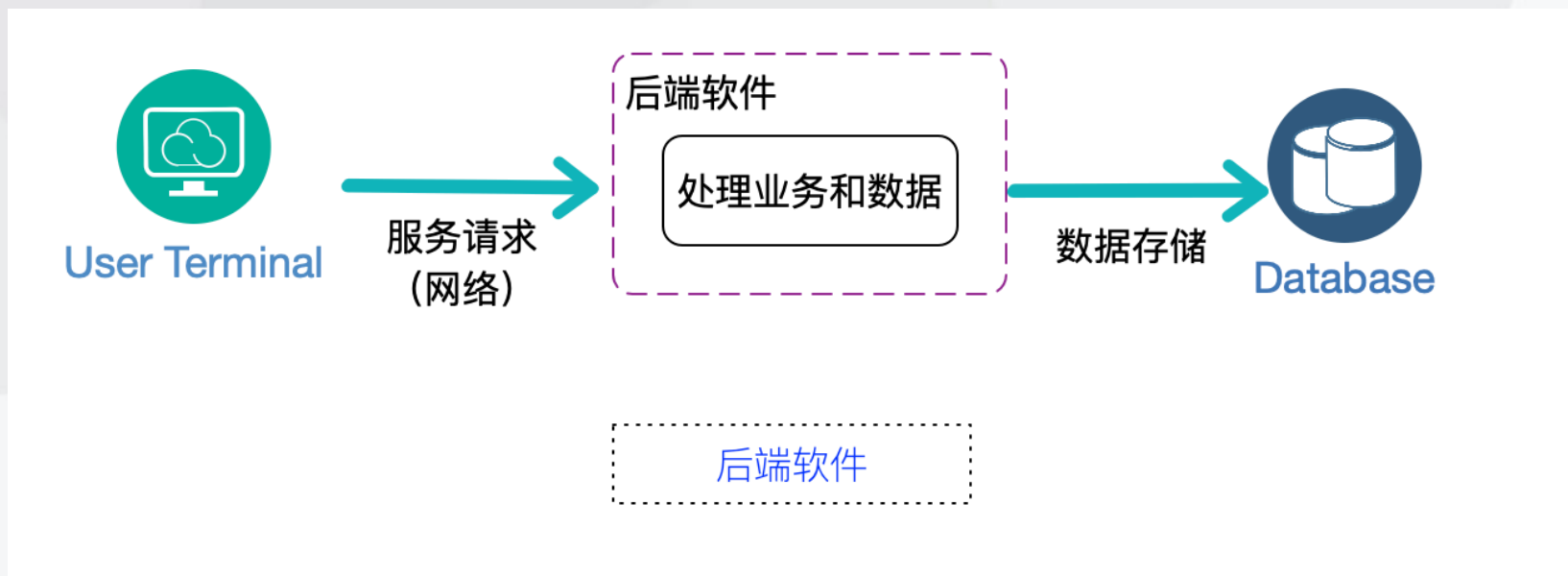


为什么使用框架

有利于程序员把控软件复杂度

有利于设计可复用的软件模块

关于后端系统的设计关注点



最简单的形态：用户少、数据少、业务少、响应要求低、需求变化小、数据聚合要求低、可以停机更新

关于后端系统的设计关注点

当我们面对各种不同情况时，对后端的软件结构就需要不同的技术来支撑

要求	用户数量	数据量	业务数量	响应要求	数据聚合	需求变化	运维要求
低	<1000	单表<100K	<100接口	0~3s	简单统计	瀑布模型处理	可以停机
中	1K~1000K	100K~100M	100~1000接口	0~3s，报表/统计 非实时响应	复杂统计、全 数据搜索	快速迭代	24*7
高	>1M	>100M	>1000接口	0~3s，所有操作	流式数据处理、 数据挖掘、数 据关系图	敏捷开发	24*7、灰度发 布、在线修复

* 这个不是标准，仅仅是我课程中方便论述

后端开发的常见高级需求~如何处理海量数据

当我们考虑“如何处理海量数据”的问题时，其实包括了两个关键

海量数据 + 处理要求

- 1) 海量数据：一般如果数据量超过1000万条我们就需要谨慎处理
- 2) 处理要求：历史数据是否参与日常业务？查询？聚合？频繁更新？

常用的一些方法

- 1) 实时业务库 + 归档库
- 2) 分库（切分）
- 3) 使用数据库集群
- 4) 旁路消息队列 + 中间结果计算
- 5) 启用特定数据库（比如启用ElasticSearch支持检索与聚合）

后端开发的常见高级需求~如何应对高并发


应对高并发是非常困难的问题，而且高并发和整个软件结构的每一个环节都有关系，所以还需要很好的区分主要的瓶颈所在的位置。经常为了应对高并发所需要关注的技术剖面：

- 1) 业务入口：采用复杂均衡设备（硬件 + 软件），配合API网关进行分发；这个同时要求了后端的服​​务是无状态的；
- 2) 业务处理：微服务网络 + 容器；可以根据服务负载动态增加服务节点提升服务能力；
- 3) 数据保存与更新：
 - 缓存 + 连接池
 - 消息队列
 - 直接使用大数据平台
- 4) 数据检索与聚合：使用大数据平台

后端开发的常见高级需求~持续运维能力

当前无论是互联网服务和物联网服务，24*7不停机要求越来越多，如何对整个系统进行诊断、问题排查，自动恢复，灰度部署，在线升级等，对软件结构设计提出了极高的要求。主要的解决方案涉及非常多的方面：


- 1) CI（持续集成）：保障开发-验证-部署一体化；对开发流程，支持工具，编译打包等都有要求；（比如gitlab、Jenkins等）
- 2) 微服务网络：一种解决了服务间互相调用的软件框架，方便扩展；（比如dubbo、Spring Cloud）
- 3) 监测：一种可以监管调用过程、运行性能等的框架；比如Jaeger，Prometheus
- 4) 容器：虚拟化，快速提供部署实施能力，并且还需要配合管理与编排系统来使用，疗效更佳。
比如docker + k8s



同学们的若干问题

关于框架

1. Spring Boot需要学习哪些知识？Spring Security是否重要？Spring家族这么多成员，需要使用哪些合适的框架？
2. Spring Boot与Python flask或者Django相比有什么优势？主流的后端框架是什么？
3. JavaEE和php，ASP相比优势？
4. Hibernate和MyBatis哪个更多一点（我们班和其它班不一样）？
5. 传统SSM使用是否越来越少？未来趋势会是graalVM的quarkus吗？




同学们的若干问题

关于设计

1. 前后端分离在Github中如何管理（两个repo分开）？如何规划前后端对数据校验的分工？
2. 关于前后端接口设计，定义者偏向前端还是后端？

关于后端学习

1. 后端学习要达到一个什么程度才算合格呢？
2. 除了增删改查，有什么途径开始学习高级技术？



同学们的若干问题

职业

1. 前端需要详细学习后端框架吗？前端最重要的（显得高大上）是学什么呢？
2. 我到底该学前端还是后端？前后端的主流技术哪些需要学习？
3. 是更多的学习工具框架还是扎实的学习基础，比如数据结构？当前库这么丰富的情况下，是否有必要深入理解底层实现？理论学习和实际操作哪个重要？
4. 对Java后端的需求多吗？

其他

1. 准备从事AI领域，与AI无关的哪些内容是需要学习的？



END

Pb&Lois