

**《软件项目管理》作业**



**作业（二）**

**专 业 力学-软件交叉班**

**姓 名 张泽**

**学 号 2021151470032**

**指导老师 毌攀良**

**成绩分数**

二零二四 年十二 月 二十八日

**个人作业要求：**

在《软件项目管理》课程中，我们提到了“选择”，这个“选择”从大的角度来讲主要是三个层次： 1、软件开发生命周期的选择； 2、软件架构的选择； 3、代码层面设计模式的选择。 从技术的角度，考虑到作业的延续性，在本课程提供的阅读书目中主要是针对软件架构，所有本次作业聚焦于软件架构。 作业的要求： 1、主要叙述一种软件架构； 2、讲述这个架构的应用场景； 3、这个架构的优点和缺点； 4、在使用这个架构的时候需要的技术栈； 5、现阶段哪些较为知名的系统和第三方软件、库使用了这个架构，并简单评价； 6、其他你还想讲的感受。

**作业：**

**1. 微服务架构的叙述**

微服务架构是一种将单一应用程序划分为一组小服务的架构模式，每个服务运行在其独立的进程中，并通常围绕特定的业务能力进行构建。这些服务可以通过定义良好的API进行通信，通常是HTTP RESTful API。

**2. 微服务架构的应用场景**

微服务架构适用于需要灵活扩展、独立部署和频繁更新的大型应用，特别是在云计算环境下。例如，电商平台、社交网络等需要高并发处理和快速迭代更新的场景。

**3. 微服务架构的优点和缺点**

优点：

灵活性高、易于扩展：每个微服务都能独立扩展，不会被其他服务的变动所影响，实现了真正的模块化运作

多样的技术栈选择：由于服务的独立性，每个服务可以根据需求选择最适合的技术栈，实现最优的业务实现

敏捷的迭代与部署：微服务架构支持快速迭代，且新版本的部署不会影响到其他的微服务，降低了升级带来的风险

缺点：

运维的挑战：尽管每个微服务都相对简单，但由于服务数量的增加，运维的复杂性也随之上升

网络依赖：微服务间的通信依赖网络，任何网络的不稳定都可能会对系统稳定性带来挑战

数据一致性问题：在微服务架构下，数据的一致性和完整性也是一大挑战

**4. 使用微服务架构需要的技术栈**

Node.js：使用事件驱动的非阻塞I/O，非常适合构建轻量级的微服务

Spring Boot + Spring Cloud：Spring生态系统提供了一套完整的微服务解决方案，能够帮助开发者快速构建、部署和扩展服务

Docker + Kubernetes：利用容器化技术和Kubernetes，可以实现微服务的自动部署、扩展和管理

**5. 知名系统和第三方软件、库使用微服务架构的情况**

许多知名的系统和第三方软件、库都采用了微服务架构，例如Netflix的流媒体服务、Amazon的电商服务等。这些系统通过微服务架构实现了高可用性、可扩展性和快速迭代

评价：微服务架构使得这些系统能够灵活应对不断变化的市场需求，同时保持了系统的稳定性和可靠性。

**6. 其他感受**

微服务架构作为一种现代的软件架构模式，它提供了一种灵活、可扩展的解决方案来应对复杂的业务需求。然而，它也带来了新的挑战，如服务间的通信、数据一致性和运维复杂性。因此，在选择微服务架构时，需要仔细权衡其优缺点，并根据具体的业务场景和技术能力做出决策。随着云计算和容器技术的发展，微服务架构的实施和管理将变得更加容易，它将继续在软件工程领域扮演重要角色。