

论文标题

**个人作业**

课程名称： 软件项目管理

任课老师： 毌攀良

学 院： 建筑与环境学院

专 业： 力软创新实验班

姓 名： 王上

学 号： 2022141470286

2024年12月

1. **关于微服务架构：**

微服务架构把一个大型复杂的系统拆分为多个小的、简单的服务，然后通过构建每个服务实现特定的业务功能来完成整个系统，而且各个部分之间是独立的，它们之间通过应用程序接口API进行通信，通过多个微服务的协作，实现整个系统的各个功能。

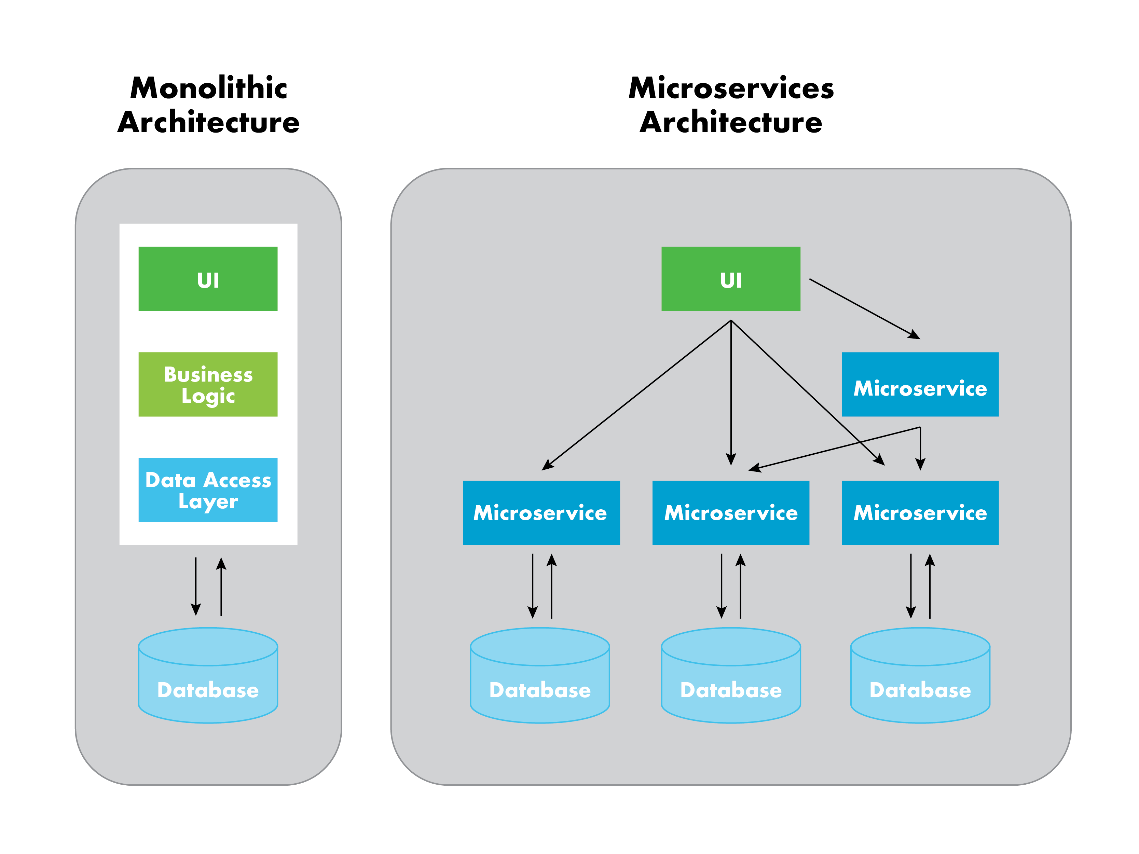


图 1单体架构与微服务架构之间的对比

不同于普通的软件架构，这种软件架构具有很好的灵活性，方便后续系统拓展、升级或修改，非常适合现在的系统开发，这种架构也适合开发团队进行分工，便于计算工作量和项目管理。同时不同的服务允许使用不同的语言和技术来开发，可以充分发挥各个技术的优势。而且每个服务都可以在不同的环境和平台中独立运行，有很好的可移植性，在一个服务发生故障时，也不会影响到整体的系统运作，具有一定的容错性。

但是这也会导致需要耗费更多的工作量去设计、测试每个小型服务，并解决各个小型服务的协同问题，以及如何提高它们之间的工作效率等问题，而且分布式的服务也会导致更高的运行维护成本。

1. **微服务架构的应用场景：**

微服务架构的核心优势在于其灵活性、可扩展性、容错性、独立部署能力和技术多样性，所以适合那些快速迭代、高度模块化的、技术多样的、高并发的系统开发。

比如对于高并发的互联网系统，可以采用微服务架构，通过增加服务器来处理更多的请求，便于根据实际用户请求量，优化升级自身系统的性能，保证在用户量激增的时候，系统仍能够保持稳定。

还有大型的企业管理系统，要牵扯到诸如签到打卡、绩效评价、日程管理、人事管理等等功能，可以使用该架构，将每种特定功能对应一个微服务，这些服务之间通过API进行协作，并且可以共用一个员工信息数据库，而且如果后续需要加入新的功能，可以很方便地添加。同时一个微服务产生故障，不会影响到整个系统，容错率比较高。

如果所做的项目需要多个团队协同合作，也适合使用微服务架构，因为微服务架构允许不同的微服务之间使用不同的开发语言和环境，方便各个开发团队独立开发与部署，而且可拓展性强，方便逐步交付。

总的来说，微服务架构的应用场景主要出现在大型企业应用、互联网高流量服务、金融、云计算和物联网等需要高度灵活性和可维护性比较强的复杂业务场景。

1. **微服务架构的优点缺点：**

微服务架构的优点：

（1）允许独立开发部署：每个微服务都可以独立部署或扩展，减少了不同服务之间的依赖关系，提高了系统的灵活性和可扩展性。同时单个服务的崩溃不会影响到整个系统。

（2）易于扩展：微服务能够水平扩展，系统规模增大时，能通过增加更多的微服务实例来保证服务，这种扩展能力使得系统能够应对不断增长的用户需求和业务挑战。

（3）便于团队协作：每个服务都可以独立开发、测试和部署，允许团队并行的工作，从而提高开发效率。

（4）持续集成和持续部署：支持频繁的代码提交和部署，如同“搭积木”的方式一样，逐步拓展系统功能，在需要快速迭代和持续交付的环境中非常有效。

微服务架构的缺点：

（1）增加了系统的复杂性：由于微服务架构将一个系统拆分成多个小型服务，所以也提升了系统的复杂程度，因此对开发团队的技术架构能力有一定要求。

（2）增加了运行时维护的成本和难度：微服务架构中存在大量的服务和依赖关系，增加了系统的运维复杂性，要求维护人员关注更多的方面，而且服务之间的通信也增加了系统的额外开销，还可能导致系统出现延迟的情况。

（3）对前期架构有更高要求：虽然微服务架构可以方便的新增功能，但是在开发初期也要提前做好规划，将系统可能涉及到的服务进行合理划分，服务划分过细可能导致服务间通信频繁、性能下降；服务划分过粗则可能失去微服务架构的优势，所以需要前期的架构准确划分各项服务。

（4）在测试和调试方面更加复杂：微服务架构中存在大量的服务和依赖关系，在测试阶段需要进行跨服务的集成测试、服务间的接口测试和各服务性能测试，而且若系统出现问题，排查和定位问题也更困难。

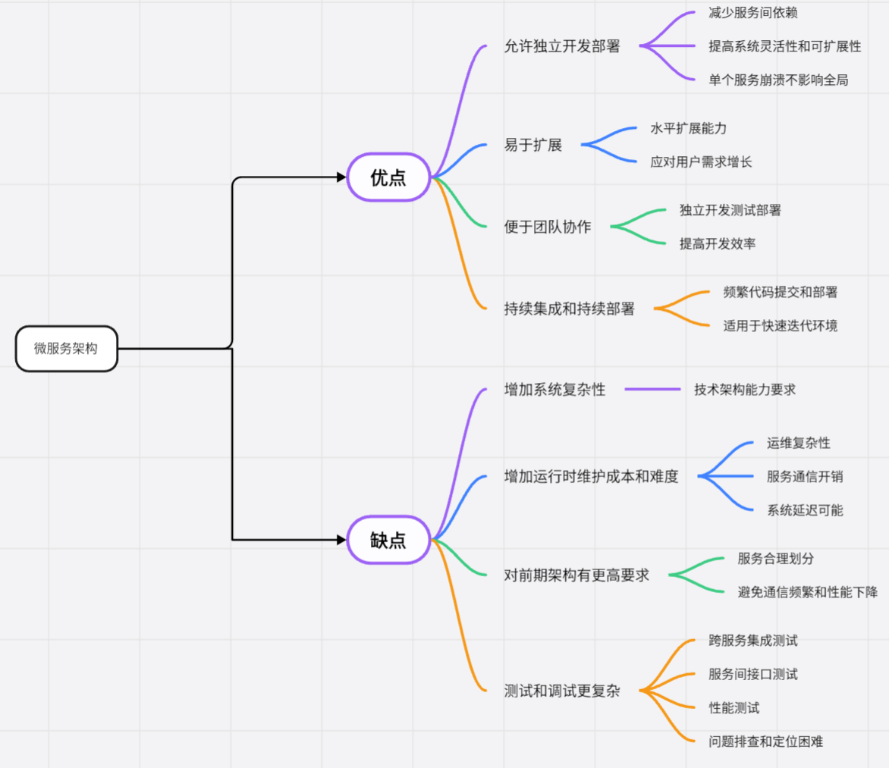


图 2介绍微服务架构的优缺点思维导图

1. **在使用微服务架构时需要的技术栈：**

微服务架构所需要的技术栈有：

（1）可以使用一些现有的开发框架，如Spring Boot等，这些开发框架简化开发和构建的过程，可以有效提高效率，快速部署微服务。

（2）由于微服务架构由不同的微服务之间协作来完成系统功能，所以需要掌握API网关来保证能通过路由功能将请求转发到目标服务。有许多的API网关可供选择，如Spring Cloud Gateway，就是一种基于Spring框架的API网关，提供流量控制等功能进行路由转发。

（3）还需掌握容器化和编排技术， 容器化可以确保系统方便的跨平台部署，并能方便拓展，在部分服务发生故障时，还可以有效隔绝故障，防止影响到整个系统。

（4）由于微服务的架构，不同微服务之间都是独立运行，但是这些服务的操作，都是作用于同一个数据库，所以要掌握数据库相关知识。并能够使用微服务来调用、修改、增删数据库信息。

1. **成功应用了微服务架构的案例：**

比较有名的有Java的基础框架spring boot，因为具有简化配置、易于部署的特点，就可以用于构建微服务架构。像国内首个基于Spring Cloud的微服务化开发平台Cloud-Platform，就是有多个模块，可以让开发者按模块逐步开发。网址：<https://gitee.com/geek_qi/cloud-platform>，还有microservices-platform，也是基于spring boot和Spring Cloud的企业级系统架构，网址如下：<https://gitee.com/zlt2000/microservices-platform>，这些都是比较知名的开源项目。

比较成功的软件案例有视频软件Netflix，也就是网飞，也是使用了微服务架构，网飞将原有的单体架构，拆分成了多个小型、松散耦合的服务。并将系统分为客户端、分发网络、数据库三个部分，并在三个部分中，通过多个微服务之间通信来实现特定的功能，分发给用户需要的内容。然后通过这些微服务之间互相协作，使得系统可以快速稳定的交付视频等信息。

同时亚马逊也是采用的微服务架构，使用AWS来拆分功能构建微服务架构，如计算服务、存储服务、网络服务等，然后为每个微服务创建独立的代码库，并使用容器化部署。因此可以保证其网络购物平台可以应对大流量访问。

1. **关于微服务架构的个人感受：**

由于现在的软件开发过程，大都倾向于“模块化”开发，因此使用微服务架构可以大展身手。将某个系统拆分成多个实现特定功能的“模块”，随后测试并部署的微服务架构，符合现有的编程趋势，是未来不可或缺的编程方式。而且其便于拓展的特点，也能够快速响应市场变化和业务需求。

虽然微服务架构也有缺点，如对通信的额外开销、增加了系统的复杂性等。但不可否认的是，微服务架构仍是一种极其有效的架构。在使用开发的时候，需要事先评估好系统的特点及要求，这样使用微服务架构可以事半功倍。但是不适合微服务架构的，使用微服务架构会带来不必要的开销。