在软件行业，微服务架构是一种重要的发展趋势。这一趋势，不仅仅是对企业内的IT信息系统建设，甚至在企业向数字化转型方面，都有着深远的影响。微服务架构与传统的单体软件架构代表着IT产业处理软件开发方式的一个根本性转变，Netflix、Google、亚马逊等组织均已成功采用这一转变。但是，与传统的单体架构相比，微服务的优势是什么呢？

1） 微服务（microservice）架构vs单体(monolithic)架构

首先，让我们来看下微服务架构和单体架构。单体应用是按单个应用程序单元来构建的。一般来说，企业内的应用程序由三部分组成：数据库（通常由关系数据库管理系统中的许多表组成），客户端用户界面（由HTML页面和/或在浏览器中运行的JavaScript组成）以及服务器端应用程序。服务器端应用程序可以处理HTTP请求，执行某些特定域的逻辑，从数据库中检索和更新数据，以及填充要发送到浏览器的HTML视图。它是一个整体——实现单个逻辑的可执行文件。如果想要对系统进行任何更改，开发人员必须构建和部署服务器端应用程序的更新版本。

相比之下，微服务通过面向业务的API接口来表达其功能。它们封装了核心的业务功能，是业务的宝贵资产。服务的实现细节（可能涉及与数据系统的集成）被完全隐藏，因为API是纯粹使用业务术语来定义的。作为业务的宝贵资产，服务可以较好地适应于多个不同的业务场景中。在业务需要的时候，同一个服务可以在多个业务流程中重用，也可以在不同的业务渠道中使用。采用松耦合的设计原则，可以最大限度地减少服务与其消费者之间的依赖关系。通过标准化的业务API表达的契约，消费者不会受到服务内部实现变化的影响。这也就允许服务的所有者可以自由实现并更改可能位于API后面的数据处理或者组合服务系统，并在不对下游的API消费者产生任何影响的情况下替换它们。

2） 使用微服务架构vs单体架构的软件开发流程

在传统的软件开发流程中（瀑布，敏捷等），通常较大规模的团队围绕一个单体应用工作。项目经理、开发人员和操作人员可以通过这些模型取得不同程度的成功，从而发布可由业务验证的候选应用程序，特别是当他们获得使用特定的软件开发和运维技术栈的经验时。然而，传统方法存在一些潜在的问题：

·单体应用可能会演变为“大泥球”，巨大又复杂;在这种情况下，很难有单个开发人员（或开发人员组）理解整个应用程序。

·单体应用很难实现模块的重用。

·扩展单体应用通常是一项较大的挑战。

·很难快速重复部署单体应用程序的更新版本。

·根据定义，单体应用是使用单个开发技术栈（即JEE或.NET）实现的，这可能会限制“为不同的任务选择正确的工具”的灵活性。

将微服务架构与云部署技术、API管理和集成技术相结合，可以为软件开发提供不同的方法。把传统模式下的单体应用拆分成独立的服务，从而可以单独开发、单独部署、单独维护。这些服务具有以下优点：

·服务粒度小，理想情况下由少数开发人员构建。

·如果公开微服务的接口使用标准协议（例如RESTful API），那么它们可以被其他服务和应用程序使用和重用，而无需通过语言绑定或共享库直接耦合。

·服务可独立部署，并且可以独立于其他服务进行扩展。

·独立地开发服务允许开发人员使用适当的开发框架来完成手头的任务。

3）微服务vs单体架构的优点

微服务架构与传统的单体架构带来的商业利益是显著的。如果部署得当，基于微服务的架构可以帮助业务避免欠下技术债务，以及大幅提高效率的重大价值。

例如，传统DevOps中，来自单体代码库的技术债务是真实存在的。使用单体代码库，即使是隔离的组件也共享相同的内存，并且共享对程序本身的访问。虽然这可能使代码接口和实现应用程序变得更容易，但它最终会削弱敏捷开发过程的灵活性。

更重要的是，单体代码库会导致效率呈指数级下降，从而增加了技术债务。例如，错误解析，界面修改，添加功能和对应用程序的其他更改等杂务会影响整个应用程序，从而造成停机，以及创建无意中引入低效率的环境。简而言之，单体代码库使用起来更耗时，适应性较差，最终维护成本更高，从而增加了技术债务。

微服务架构减少了传统的单体架构带来的技术债务，为市场节约可观的时间和速度成本，这是其一项重要优势。另外，微服务架构的优势不仅仅只有这一点，它还为企业带来了其他好处，从而可以降低成本并提高利润。这些好处包括以下几点：

·敏捷性：通过将功能分解到最基本的级别然后抽象相关服务，DevOps可以只专注于更新应用程序的相关部分。这消除了通常与单体应用程序相关的痛苦的集成过程。微服务加速了开发，将其转变为可在数周而非数月内完成的流程。

·效率：微服务架构可以更有效地使用代码和底层基础设施。通过减少运行特定应用程序所需的基础架构数量，可以节省多达50％的成本，这种情况并不少见。

·弹性：通过在多个服务之间分散功能，可以消除应用程序对单点故障的敏感性。从而使应用程序能够更好地运行，减少停机时间并可按需扩展。

·收益：更快的迭代和更短的停机时间可以帮助增加收益。随着微服务的不断改进，用户保留和参与度也会提高。

公司如果希望最大限度地提高生产力，提高敏捷性和改善客户体验，那么就应该从采用单体Web应用，改为采用微服务，其松耦合的架构可加速开发，测试和部署，从而满足当今和未来的数字需求。

4）实现微服务的相关技术

Docker技术， K8S技术， Resfful API, Cloud技术。

5）