

漫画：什么是微服务？

2018-01-15 11:19:19 分类：Web开发 (/category/WebDev1024/)

来自：程序员小灰 (<https://mp.weixin.qq.com/s/FRVOYlgZCO524KwzQRohLA>) (微信号：chengxuyuanxiaohui)，作者：玻璃猫

小灰，你最近脸色不太好啊，
出了什么事？



哎，还不是被我那个猪队友给
连累的……



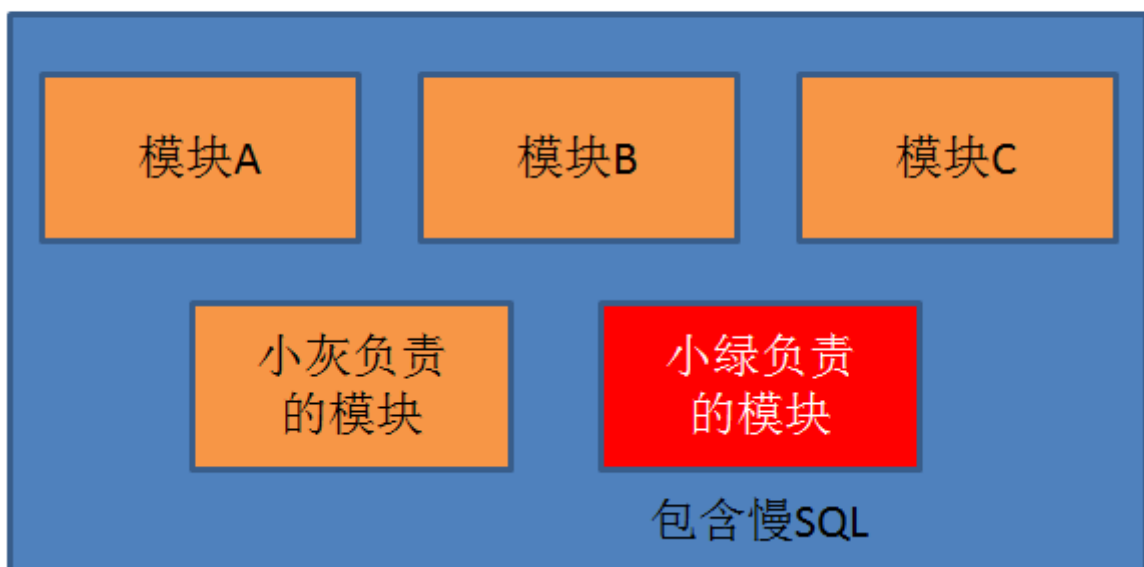
哦？什么猪队友，说来听听？



昨天晚上，我的同事小绿上线的代码里有一条慢 sql，结果……



小灰他们公司的某系统



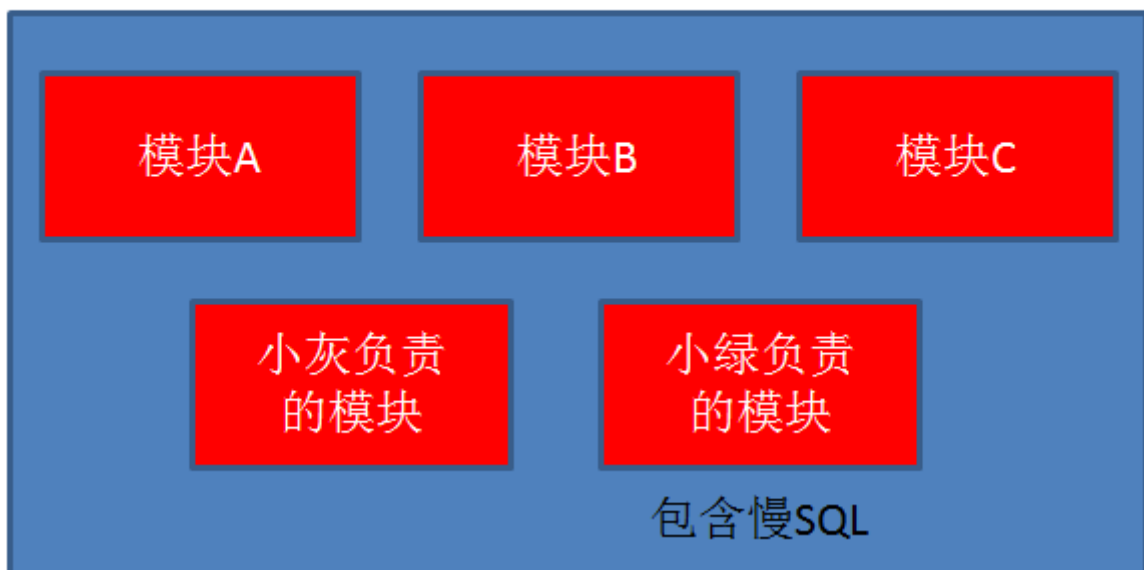


连接池都打满了，整个系统

一起挂掉。我们查了半天才查到问题并且修复好，老板把我们大骂了一顿。



小灰他们公司的某系统



(/)

嘿嘿，如果我是你们老板，
我也得骂你们！



啊，为什么呀？

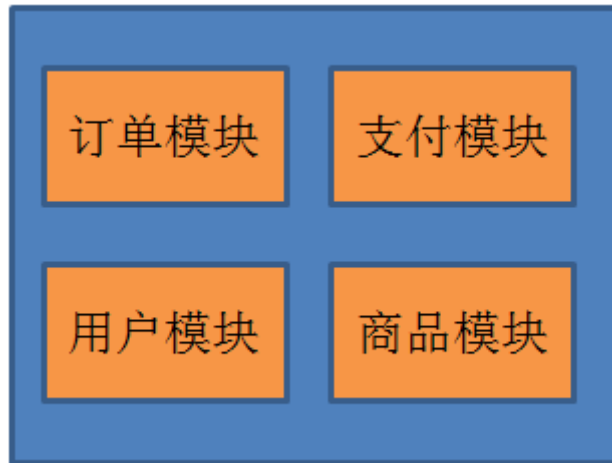


单体架构的痛点

你们现在的系统架构属于传统的
MVC 架构，所有业务子模块都集成
在一个很重的 JVM 进程当中。



(/)



这种单体架构的好处是便于管理，所有代码都在同一个项目当中。但是当产品规模越来越大，其坏处也很明显：



缺点一：项目过于臃肿

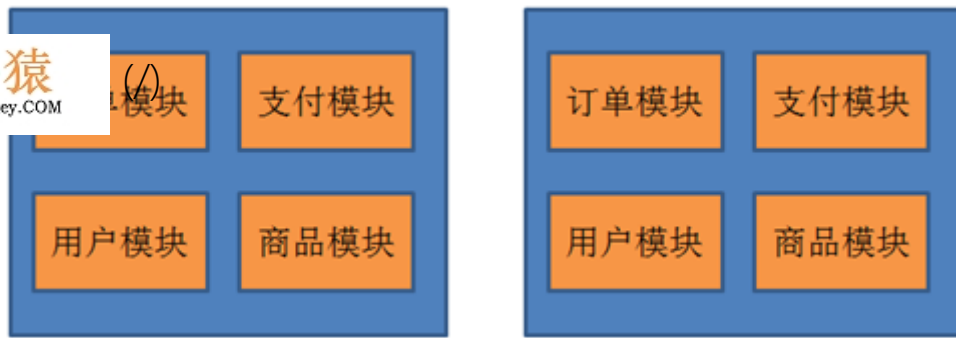
当大大小小的功能模块都集中在同一项目的时候，整个项目必然会变得臃肿，让开发者难以维护。

缺点二：资源无法隔离

就像刚刚小灰的经历一样，整个单体系统的各个功能模块都依赖于同样的数据库、内存等资源，一旦某个功能模块对资源使用不当，整个系统都会被拖垮。

缺点三：无法灵活扩展

当系统的访问量越来越大的时候，单体系统固然可以进行水平扩展，部署在多台机器上组成集群：



但是这种扩展并非灵活的扩展。比如我们现在的性能瓶颈是支付模块，希望只针对支付模块做水平扩展，这一点在单体系统是做不到的。

单体架构存在这么多问题……

要怎么来解决呢？



办法倒是有一个，我们可以把臃肿的单体系统拆分成「微服务」。



什么是微服务？



微服务（Microservice Architecture）是近几年流行的一种架构思想，关于它的概

究竟什么是微服务呢？我们在此引用 ThoughtWorks 公司的首席科学家 Martin Fowler 的一段话：



In short, the microservice architectural style is an approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms, often an HTTP resource API. These services are built around business capabilities and independently deployable by fully automated deployment machinery. There is a bare minimum of centralized management of these services, which may be written in different programming languages and use different data storage technologies.

谷歌翻译如下：

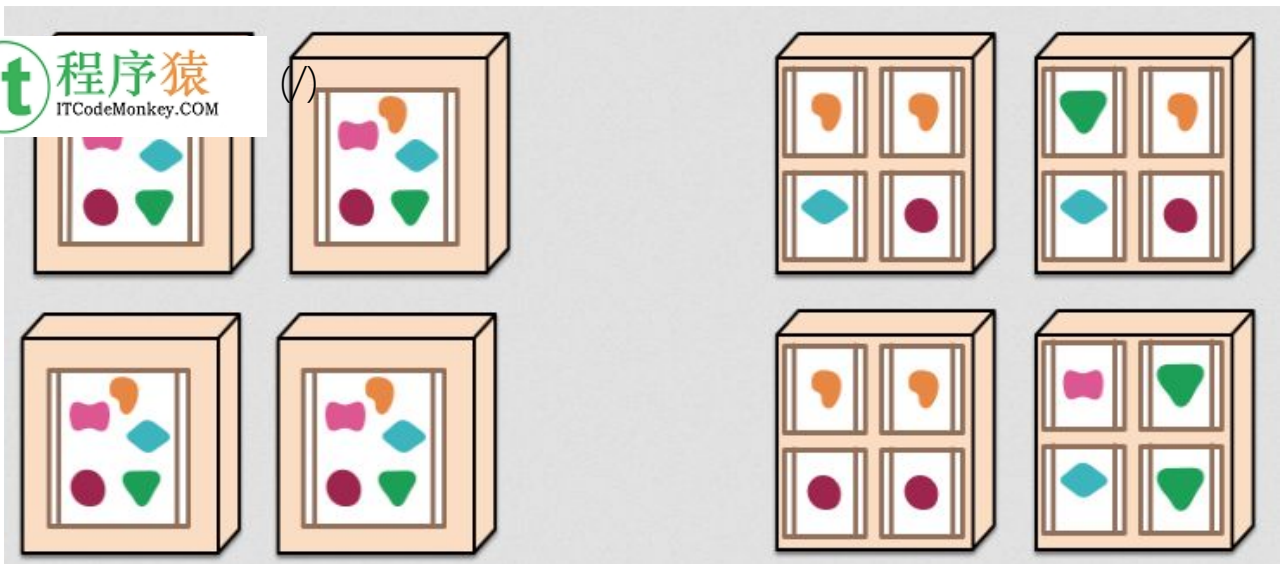
简而言之，微服务架构风格是一种将单个应用程序作为一套小型服务开发的方法，每种应用程序都在自己的进程中运行，并与轻量级机制（通常是HTTP资源API）进行通信。这些服务是围绕业务功能构建的，可以通过全自动部署机制独立部署。这些服务的集中管理最少，可以用不同的编程语言编写，并使用不同的数据存储技术。

说了这么多概念，微服务有什么样的具体特点呢？

1、独立部署，灵活扩展

传统的单体架构是以整个系统为单位进行部署，而微服务则是以每一个独立组件（例如用户服务，商品服务）为单位进行部署。

用一张经典的图来表现，就是下面这个样子：



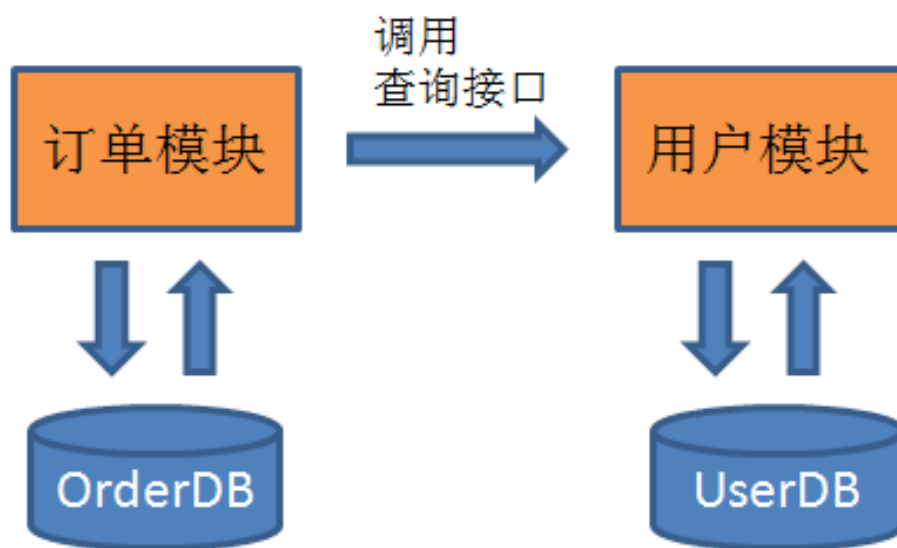
图中左边是单体架构的集群，右边是微服务集群。

什么意思呢？比如根据每个服务的吞吐量不同，支付服务需要部署20台机器，用户服务需要部署30台机器，而商品服务只需要部署10台机器。这种灵活部署只有微服务架构才能实现。

而近几年流行的**Docker**，为微服务架构提供了有效的容器。

2、资源的有效隔离

微服务设计的原则之一，就是每一个微服务拥有独立的数据源，假如微服务A想要读写微服务B的数据库，只能调用微服务B对外暴露的接口来完成。这样有效避免了服务之间争用数据库和缓存资源所带来的问题。



同时，由于每一个微服务实例在**Docker**容器上运行，实现了服务器资源（内存、CPU资源等）的有效隔离。

3、团队组织架构的调整



有测试的团队。

也改变了原有的企业研发团队组织架构。传统的研发组织架构有前端的团队，后端有后端的团队，DBA有DBA的团队，测试

水平团队组织架构



而微服务的设计思想对团队的划分有着一定的影响，使得团队组织架构的划分更倾向于垂直架构，比如用户业务是一个团队来负责，支付业务是一个团队来负责。

2018-01-15

当然，这种垂直划分只是一个理想的架构，**实际在企业中并不会把团队组织架构拆分得这么绝对。**

微服务与面向服务架构SOA的区别



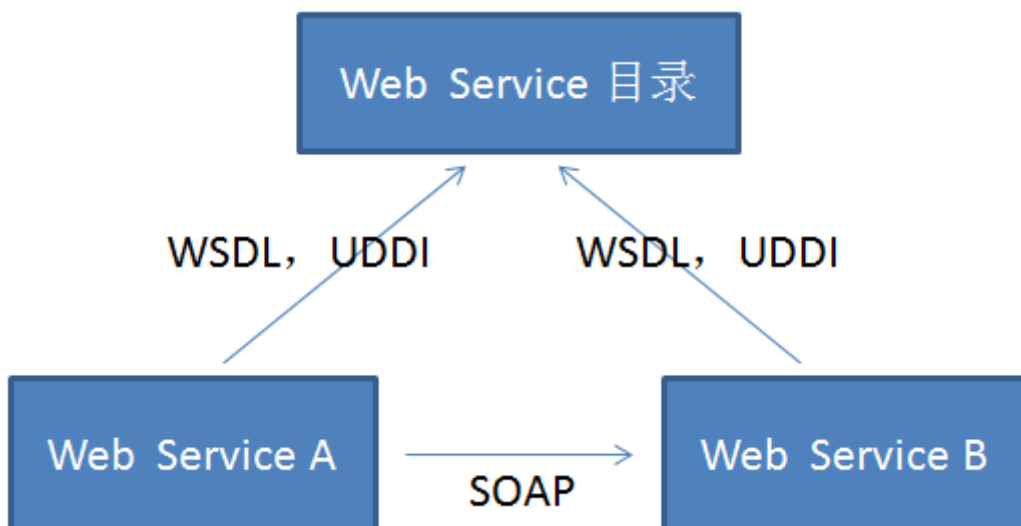
还有一个疑问，这个微服务来和 SOA 架构思想好像差不多，它们之间有什么不同呢？



这个问题提的很好，SOA 架构是一种粗粒度、松耦合的服务架构，其更多的是强调异构系统之间的服务通信。



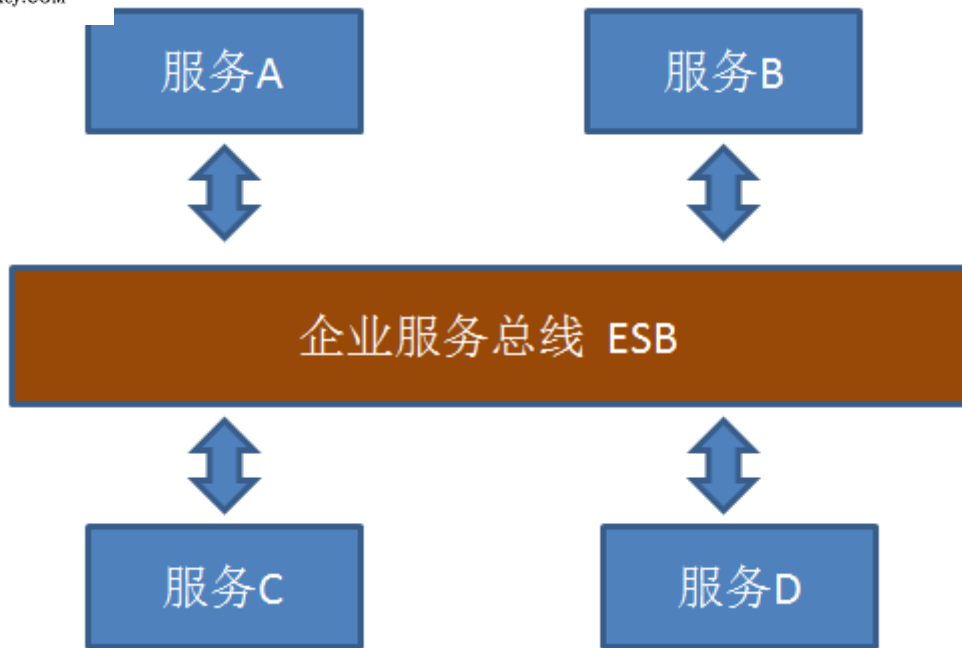
SOA是什么样子的呢？可以是下面这样的Web Service：



也可以是下面这样的ESB企业服务总线：



(/)



总之，SOA架构强调的是异构系统之间的通信和解耦合，而微服务架构强调的是系统按业务边界做细粒度的拆分和部署。

那么，我可以认为使用 Dubbo 框架的项目是 SOA 架构，使用 Spring Cloud 框架的项目是微服务架构吗？



不不不，无论 Dubbo 还是 Spring Cloud 都是框架和组件，SOA 和微服务是架构思想，它们不是同一个层面的概念。



我们可以说 Dubbo 和 Spring Cloud 很好地支持了 SOA 和微服务架构，但不能说 Dubbo 是 SOA，Spring Cloud 是微服务。



微服务架构的不足

最后一个问题，微服务架构有什么样的缺点呢？



说起缺点，一方面微服务把原有的项目拆成多个独立工程，增加了开发和测试的复杂度。



我记得以前公司的某个项目就把服务拆分得过细，导致添加一个小功能需要修改四五个工程，这显然是糟糕的设计。



再有就是微服务架构需要保证不同服务之间的数据一致性，引入了分布式事务和异步补偿机制，为设计和开发带来一定挑战。



所以说架构设计没有绝对的好与坏，
关键还要看应用场景。微服务的思想
我们就介绍到这里，感谢大家的支持！



来自：程序员小灰（微信号：chengxuyuanxiaohui），作者：小灰



推荐↓↓↓



Web开发

上一篇：HTTPS 通俗演义：爱丽丝与信鸽的故事 (/article/1877.html)

下一篇：前端性能优化 —— 移动端浏览器优化策略 (/article/1915.html)



(/)

1 在阿里工作几年，女友跑路，晋升无望，买不起房，我活成了一个笑话~ (/article/15942.html)

2 漫话：如何给女朋友解释什么是“大案牍术”？ (/article/15938.html)

3 抓狂！这条Linux命令执行完女朋友都跟人跑了！ (/article/15937.html)

4 7921 Star！Python学习的必备法宝，随查随用，太方便了吧 (/article/15921.html)

5 安全研究人员最爱的12款Linux发行版 (/article/15919.html)

© 2017-2018 IT程序猿 闽ICP备08108865号-1