**停止盲目使用微服务**

<https://mp.weixin.qq.com/s/MAGTPVBxcZOCBOKffxKLBA>

**管理数据是一场噩梦**

保持微服务间的数据同步可能是一项挑战。

每个微服务都有一个数据库，这是推荐的模式。它允许松散的耦合，并且可以让特定服务团队在无需放慢速度协作共享代码的情况下，独立地工作。但如果本应同步启动的微服务中的一个出现故障时，会发生什么呢？比如，其中一个微服务更新了其数据库，而另外一个却没有。这种情形会导致数据不一致。

根据个人的经验，调查跨服务的数据不一致会非常痛苦。错误的跨服务性质需要一个人在不同的服务中工作来修正错误。遗憾的是，这就导致了微服务的优势，即专门针对团队的服务，无法发挥作用。

在一个单体应用中，只要把两个数据库调用合并到一个原子事务中，就能很容易地避免这种情况，因此，所有的插入都会成功，或者都不会成功。非常的简单。但是，松散的藕合会使微服务变得更为难以实现。

### 设置时间更长

构建一个微服务架构所花费的时间要比将相同的特性整合到一个单体应用中要多得多。尽管单个服务是非常简单的，但是交互的服务集合要远比单一的单体更加复杂。在一个单体中，一个函数可以调用任何其他公共函数。但是，微服务中的函数仅限于调用同一个微服务中的函数。这就需要服务之间的通信。构建 API 或者消息传递来促进这一点并不容易。而且，跨微服务的代码重复也是不可避免的。当一个单体应用可以一次定义一个模块并多次导入，而微服务是它自己的应用：在每一个模块都必须定义模块和库。

### 微服务最适合大型团队

将微服务分派到各个团队的奢侈做法是留给大型工程部门。尽管这对这个架构来说是一个很大的优势，但是如果你拥有足够的工程师来为每一项服务指定一些工程师，那么这才是可行的。减少代码范围，可以让开发人员对代码有更好的理解，加快开发的速度。但是，大部分的初创公司都没有这样的奢侈。在一个创业早期的公司，由于缺乏足够的资源，有些工程师必须在所有的服务之间工作。遗憾的是，这样做会降低工作效率，因为在不同的应用中跳跃，可能会导致环境的变化。我发现，在我已经很久没有关注的微服务中调查 Bug，是一件非常令人筋疲力尽的事情。

### DevOps 更复杂

选择微服务最有说服力的一个原因就是可以在不同类型的服务器上运行不同的服务。这是为什么呢？React 前端的内存、CPU 和启动时间的需求与训练机器学习模型的服务大相径庭。为每一项服务选择适当的基础架构类型，可以极大地减少费用。但是，这也给自己带来了一个挑战。

举个例子，在我的职业生涯初期，由于忘记重启一个更新过代码的服务，导致我丢失了大量的生产数据。过期的代码会通过 API 来接收数据，却没有把数据存入数据库，反而消无声息地失败。这样的数据就会永远丢失了。

我之所以提出这一点，是想要表明，配置、维护和监控多个微服务，要比单一的单体应用要复杂得多。拥有多个应用程序，还为骇客增加了多个攻击面。

从理论上讲，“松散耦合”的服务允许每个服务在其他服务失败时继续工作。但这只是一厢情愿的想法：对于有客户的复杂业务来说，很难实现真正的松散耦合。

最终，你的应用程序架构的可靠程度取决于最薄弱的部分。移动的碎片越多，出错的可能性就越大。

### 总结

许多公司使用微服务并不是真正需要它们，而且尽管微服务现在很流行，但它们并不适合初学者。大多数公司最好的做法是构建一个单体，然后在绝对必要的时候将单体的部分拆分到微服务中。

把从头开始的微服务架构的机会留给那些财力雄厚的大型科技公司。

你的早期阶段的创业公司也许还没有准备好。我的公司就没有准备好，结果，让我们付出了大量的时间和精力。