

02 | 从单体应用走向服务化

2018-08-25 胡忠想

从0开始学微服务

[进入课程 >](#)



讲述：胡忠想

时长 07:55 大小 3.64M



专栏上一期，我给你讲述了什么是微服务，以及微服务架构的由来。简单回顾一下，微服务就是将庞杂臃肿的单体应用拆分成细粒度的服务，独立部署，并交给各个中小团队来负责开发、测试、上线和运维整个生命周期。

那么到底什么时候应该拆分单体应用？拆分单体应用有哪些标准可依呢？

为了解答这两个问题，今天我将通过具体案例来阐述，希望你能够学会**单体应用拆分成微服务的正确姿势**。

什么时候进行服务化拆分？

从我所经历过的多个项目来看，项目第一阶段的主要目标是快速开发和验证想法，证明产品思路是否可行。这个阶段功能设计一般不会太复杂，开发采取快速迭代的方式，架构也不适合过度设计。所以将所有功能打包部署在一起，集中地进行开发、测试和运维，对于项目起步阶段，是最高效也是最节省成本的方式。当可行性验证通过，功能进一步迭代，就可以加入越来越多的新特性。

比如做一个社交 App，初期为了快速上线，验证可行性，可以只开发首页信息流、评论等基本功能。产品上线后，经过一段时间的运营，用户开始逐步增多，可行性验证通过，下一阶段就需要进一步增加更多的新特性来吸引更多的目标用户，比如再给这个社交 App 添加个人主页显示、消息通知等功能。

一般情况下，这个时候就需要大规模地扩张开发人员，以支撑多个功能的开发。如果这个时候继续采用单体应用架构，多个功能模块混杂在一起开发、测试和部署的话，就会导致不同功能之间相互影响，一次打包部署需要所有的功能都测试 OK 才能上线。

不仅如此，多个功能模块混部在一起，对线上服务的稳定性也是个巨大的挑战。比如 A 开发的一个功能由于代码编写考虑不够全面，上线后产生了内存泄漏，运行一段时间后进程异常退出，那么部署在这个服务池中的所有功能都不可访问。一个经典的案例就是，曾经有一个视频 App，因为短时间内某个付费视频访问量巨大，超过了服务器的承载能力，造成了这个视频无法访问。不幸的是，这个网站付费视频和免费视频的服务部署在一起，也波及了免费视频，几乎全站崩溃。

根据我的实际项目经验，一旦单体应用同时进行开发的人员超过 10 人，就会遇到上面的问题，这个时候就该考虑进行服务化拆分了。

服务化拆分的两种姿势

那么服务化拆分具体该如何实施呢？一个最有效的手段就是将不同的功能模块服务化，独立部署和运维。以前面提到的社交 App 为例，你可以认为首页信息流是一个服务，评论是一个服务，消息通知是一个服务，个人主页也是一个服务。

这种服务化拆分方式是**纵向拆分**，是从业务维度进行拆分。标准是按照业务的关联程度来决定，关联比较密切的业务适合拆分为一个微服务，而功能相对比较独立的业务适合单独拆分为一个微服务。

还有一种服务化拆分方式是**横向拆分**，是从公共且独立功能维度拆分。标准是按照是否有公共的被多个其他服务调用，且依赖的资源独立不与其他业务耦合。

继续以前面提到的社交 App 举例，无论是首页信息流、评论、消息箱还是个人主页，都需要显示用户的昵称。假如用户的昵称功能有产品需求的变更，你需要上线几乎所有的服务，这个成本就有点高了。显而易见，如果我把用户的昵称功能单独部署成一个独立的服务，那么有什么变更我只需要上线这个服务即可，其他服务不受影响，开发和上线成本就大大降低了。

服务化拆分的前置条件

一般情况下，业务系统引入新技术就必然会带来架构的复杂度提升，在具体决策前，你先要认识到新架构会带来哪些新的问题，这些问题你和你的团队是否能够解决？如何解决？是自己投入人力建设，还是采用业界开源方案？

下面几个问题，是从单体应用迁移到微服务架构时必将面临也必须解决的。

服务如何定义。对于单体应用来说，不同功能模块之前相互交互时，通常是以类库的方式来提供各个模块的功能。对于微服务来说，每个服务都运行在各自的进程之中，应该以何种形式向外界传达自己的信息呢？答案就是接口，无论采用哪种通讯协议，是 HTTP 还是 RPC，服务之间的调用都通过接口描述来约定，约定内容包括接口名、接口参数以及接口返回值。

服务如何发布和订阅。单体应用由于部署在同一个 WAR 包里，接口之间的调用属于进程内的调用。而拆分为微服务独立部署后，服务提供者该如何对外暴露自己的地址，服务调用者该如何查询所需要调用的服务的地址呢？这个时候你就需要一个类似登记处的地方，能够记录每个服务提供者的地址以供服务调用者查询，在微服务架构里，这个地方就是注册中心。

服务如何监控。通常对于一个服务，我们最关心的是 QPS（调用量）、AvgTime（平均耗时）以及 P999（99.9% 的请求性能在多少毫秒以内）这些指标。这时候你就需要一种通用的监控方案，能够覆盖业务埋点、数据收集、数据处理，最后到数据展示的全链路功能。

服务如何治理。可以想象，拆分为微服务架构后，服务的数量变多了，依赖关系也变复杂了。比如一个服务的性能有问题时，依赖的服务都势必会受到影响。可以设定一个调用性能阈值，如果一段时间内一直超过这个值，那么依赖服务的调用可以直接返回，这就是熔断，也是服务治理最常用的手段之一。

故障如何定位。在单体应用拆分为微服务之后，一次用户调用可能依赖多个服务，每个服务又部署在不同的节点上，如果用户调用出现问题，你需要有一种解决方案能够将一次用户请求进行标记，并在多个依赖的服务系统中继续传递，以便串联所有路径，从而进行故障定位。

针对上述问题，你必须要有可行的解决方案之后，才能进一步进行服务化拆分。专栏后面的文章，我会给你逐一讲解相应的解决方案。

总结

无论是纵向拆分还是横向拆分，都是将单体应用庞杂的功能进行拆分，抽离成单独的服务部署。

但并不是说功能拆分的越细越好，过度的拆分反而会让服务数量膨胀变得难以管理，因此找到符合自己业务现状和团队人员技术水平的拆分粒度才是可取的。我建议的标准是按照每个开发人员负责不超过 3 个大的服务为标准，毕竟每个人的精力是有限的，所以在拆分微服务时，可以按照开发人员的总人数来决定。

思考题

想想你现在的业务场景，如果是单体应用的话，是否需要服务化拆分？如果需要的话，你觉得纵向拆分还是横向拆分合适？具体可以拆分到什么粒度？

欢迎你在留言区写下自己的思考，与我一起讨论。

从 0 开始学微服务

微博服务化专家的一线实战经验

胡忠想 微博技术专家



新版升级：点击「 请朋友读」，10位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 01 | 到底什么是微服务？

下一篇 03 | 初探微服务架构

精选留言 (72)

写留言



oddrock

2018-08-27

88

我现在对拆分的考量：

一是业务维度聚类，业务和数据关系密切的应该放在一起。

二是功能维度聚类，公共功能聚合为一个服务。

三是人员聚类，这是个实际中的考量，如果某几个业务就是这几个人比较熟，那么最好放在一起，未来开发部署都好办。...

展开

作者回复: 总结的很好，看得出来属于业务经验很丰富的。😊



唐伯虎点蚊...

2018-08-25

59

纵向拆分和横向拆分感觉还是不太理解

展开



日拱一卒

2018-08-26

27

关于服务拆分策略，我理解在实际工作中应该是横向和纵向相结合的方式：

1. 首先收集整理公用的模块，将其进行服务化处理，这是横向拆分。
2. 其次根据不同业务之间的耦合程度，将相对独立的服务拆分到不同的服务中，这属于纵向拆分。

...

展开

作者回复：对，实际业务中这两种拆分方式一般是相互结合使用的，如果业务比较多分散，适合做纵向拆分。如果多个业务之间有公共模块耦合，适合把公共模块拆分出来，适合做横向拆分。



snakorse

2018-08-25

15

还有必须要解决的一个问题是 拆分后带来的跨服务事务的一致性问题



jogin

2018-08-25

13

看评论感觉好多人不需要微服务。

1. 业务复杂度问题单体应用没能梳理清楚，微服务也搞不懂定。
2. 人手太少，微服务只会增加系统复杂度，运维成本，大家不要看到把原先的函数调用改成接口调用，没真正了解服务拆分后带来的系统复杂度。
3. 系统设计问题，代码写的乱，开发流程问题，不是微服务能解决的。

展开

作者回复：对，微服务只能解决单体膨胀后的痛，但也会带来架构复杂性，要权衡利弊



常玉棋

10

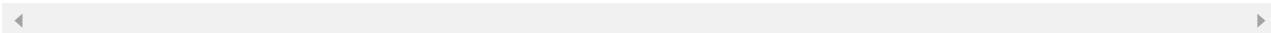


2018-08-25

我觉得目前单体能搞定的话就不要为了拆分而拆分，因为拆分后涉及到的问题有可能会让现有人员手忙脚乱，最好等做好了技术和人员准备再拆分。

展开 ▾

作者回复: 是这样的



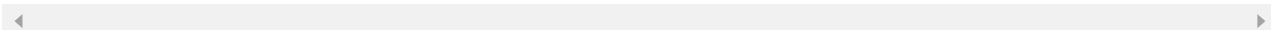
明天更美好

2018-08-25

👍 8

我们是单体应用，一共22个接口，但是有一个接口并发巨大，二期势必拆分微服务架构。我觉得按照功能进行垂直拆分比较合适，我们之前有17个开发人员，现在裁员就剩下4个。希望胡老师给个建议

作者回复: 4个人搞微服务，有没有基础架构人员？没有就算了吧.....



科大大👉...

2018-08-25

👍 7

当公司技术人员不足时就没必要考虑微服务了，好好做好单体架构，但是做的同时也要考虑未来有可能过渡到微服务，所以如果明确了以后会变成微服务的话单体架构有没有什么设计标准呢？



👉 ~ ...

2018-08-26

👍 6

横向拆分:把一些不会和其他主业务有依赖的功能拆分出来，最典型的就是common。
纵向拆分:按照业务来拆分，比如用户访问一个商城，从访问到下单成功会调用很多服务，我们可以把用户登录做一个服务，商城首页以及商品信息做一个服务，购物车模块是一个服务，支付模块又是另一个服务，这样整个商城都是微服务架构，不管哪个单独的服务出现问题，都不会影响别的服务，方便于技术人员最快定位问题，解决问题。...

展开 ▾

作者回复: 对，视具体项目而定





丢

2018-08-28

👍 5

实际项目通常是横向和纵向拆分相结合，公共的服务单独拆分出来，然后上层业务按功能模块拆分



Geek_8d2ca...

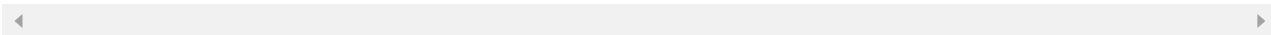
2018-08-28

👍 5

开始是10多人的团队，对业务进行了拆分，上线后，人员陆续离职，现在是6个人背着将近30个微服务，开发一个功能点，基本上所有人都得上，就演变成天天开会还解决不了问题，效率奇低.....所以，有没有微服务退回到单体的做法？

展开 ▾

作者回复: 可以适当做服务合并



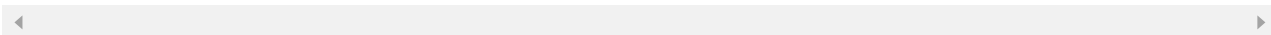
lovedebug

2018-08-28

👍 4

微服务对运维和运维架构要求很高，没有运维就搞微服务就是给自己埋坑。

作者回复: 是的



eason2017

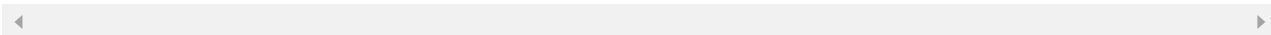
2018-08-25

👍 4

前期，可以稍微粗粒度一些。先进行纵向拆分，把基础功能（用户系统）独立部署以维护。其它业务功能关联不紧密的可以独立部署，可以看这些业务在公司发展方向的重要性后面，可以看清哪些功能和其他业务系统一定要调用的，同时，自身系统内也有其他繁杂的功能，那么可以进行横向切分，把被频繁调用的服务抽象并独立部署。

展开 ▾

作者回复: 对



少帅

2018-08-26

👍 3

是否需要拆分要看目前公司项目的具体的情况，如果现有架构无法满足当前业务发展速度，比如构建一次要半天，上一次线经常搞通宵，牵一发而动全身，那么这个时候可以考虑一下微服务，拆微服务势必会提高运维、问题追踪、分布式事务等复杂度，所以微服务整个技术栈都要考虑进去

展开 ▾

作者回复: 嗯，搞不定微服务的技术栈就不要往这个坑里跳



CN光...

2018-08-25

👍 3

目前根据业务 拆分出来了两个 独立服务 接口走rpc 调用 还在测试环境验证中！



jogin

2018-08-25

👍 3

没有推行微服务之前，单体应用的业务逻辑拆分，整体系统设计做得好，会给后期服务化改造铺平道路。



softtwilig...

2018-08-25

👍 3

横向拆分我的理解是就像aop，把共用但又跟其他部分不太相关的抽离出来，而纵向拆分是把高没聚的拆分出来

展开 ▾



先生

2018-08-27

👍 2

我们不要为了微服务而做微服务，做了服务拆分，我们会投入更大。个人觉得，做了服务拆分后，带来的最大的挑战应该是分布式事务，这个原因也导致了我们的微服务的难落地。另外的话，大部分的初创公司不适合，需要结合公司自身的人员储备和业务情况等来综合考虑。

展开 ▾

作者回复: 是的



jogin

2018-08-25

👍 2

什么时候推行微服务，不同技术栈有区别，之前最多的一次4，50个php技术维护一个单体应用，整个系统结构，业务模块之间划分比较清晰，不同小组负责不同模块，开发上采用多版本并行开发，每个版本7，8个技术。这时候整个团队开发效率上没有太大问题，架构组把控非常好。

展开 ▾



Ivan

2018-08-25

👍 2

之前在其他专栏用的是“三个火枪手”原则，即:三个人开发一个微服务。而这里的建议是一个人负责不超过三个微服务。这里不是很理解。帮忙解答一下，谢谢

展开 ▾

作者回复: 维度不一样，三个火枪手是从开发角度，我讲的是负责维护角度

