链滴

微服务设计

作者：feiwo
原文链接：https://ld246.com/article/1607863731584
来源网站：链滴
许可协议：署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)

不要把一堆鸡蛋放在同一个篮子里面。在投资的时候，我们一般不会把全部身家都放在一只股票或基上面，一般都会采取组合的形式，进行投资理财，分摊风险，降低亏损率。软件服务也是一样，如果单点服务，如果出现故障，那么整个服务也将不可用了，对整个产品造成恶劣的影响，导致用户大面流失。

一、去中心化

在微服务中，一般我们需要三个去中心化：

1. 技术去中心化
2. 数据去中心化
3. 治理去中心化

技术去中心化

在开发微服务中，每个服务面向的业务场景皆不相同，有些是数据密集型业务服务，需要大内存大存，如：内容系统。有些是计算密集型，需要良好的

CPU或GPU进行大量计算提供服务，如：AI，视频处理等。我们需要为不同的服务选择不同的技术解方案。在选择技术栈时，要避免过度多样化，需要结合团队实际的情况来取舍，要是没个服务都是用同的语言来实现，维护成本将会很高。

数据去中心化 — 服务之间通过RPC通讯

微服务中，每个服务将独享自身的数据存储设施(数据库and缓存等)，不像传统的应用共享一个

DB和Cache。这样有利于服务的独立性，隔离相关干扰。

治理去中心化

当出现流量热点时，大流量访问某一个服务，流量经过

Nginx进行转发，单个Nginx服务无法承载大量流，那么，就需要部署多套Nginx，防止单个Nginx服被大流量打死，从而导致服务不可用。一般的，遇到服务瓶颈时，要考虑纵向扩展与横向扩展的成本进行对比，选择目下最佳方案。

二、 基础设施自动化

一般来说，无自动化不微服务，一旦实行微服务架构，就意味着不再像传统的应用，始终只有很少的务，部署起来很方便。微服务一般都会拆成十几二十个(主要看应用规模)，可独立运行的服务，这意着，开发，调试，测试，监控和部署复杂度会相应的增大，必须要有合适的自动化基础设施来支持微务的架构模式。

一般基础设施(包括不限于)

1. CI/CD — Gitlab + Gitlab Hooks + k8s
2. Testing — 单元测试， API自动化测试(Yapi)
3. 在线运行时 — k8s， 以及一系列的，日志收集，日志监控，性能监控等等

三、服务可用性与兼容性

可用性

微服务架构采用粗粒度的进程间通信，引入了额外的复杂性和需要处理的新问题，如：网络延迟，消格式，负载均衡和容错。提高可用性需要关注如下几点：

1. 隔离
2. 超时控制
3. 负载保护
4. 限流
5. 降级
6. 重试
7. 负载均衡

兼容性

一单采用了微服务架构模式，那么在微服务变更时，我们需要特别小心，服务提供者的变更可能引发务消费者的兼容性被破坏，需要时刻谨记保持服务契约接口的兼容性，一般的原则是遵循**博斯塔尔法则** — 发送时保守,接收时开放，在开发时，面向失败编程，提高代码的鲁棒性。

1. 发送时保守 — 最小化传递必要的信息
2. 接收时开放 — 最大限度容忍冗余数据，保证兼容性

四、组件服务化

传统实现组件的方式是打包成一个库，如：

Java下的Jar包， Windows下的.dll，Linux下的.so，都属于一个库，集成进现有的应用，就可以使用提供的相应的功能。库和现有的应用打包成一个可执行的服务，运行在同一个进程中，这意味着库的部变化，整个应用都需要重新部署。在微服务模式下，通过微服务实现组件，将单个应用拆分为一系的服务组件，运行在不同的进程中，那么单一服务的局部变化，只需要更新部署对应的服务进程即可。

Golang实现一个微服务组成

Kit — 一个微服务的基础库(框架)， 如： go-kit, go-micro, kratos,TarsGo

Service — 业务代码 + kit依赖 + 第三方依赖，组成业务微服务

RPC + Message Queue — 轻量级通讯, 如：gRpc, Rest Api

五、服务划分

按业务组织服务 —— 按业务能力组织服务的意思是服务提供的能力和业务能力对应。譬如：

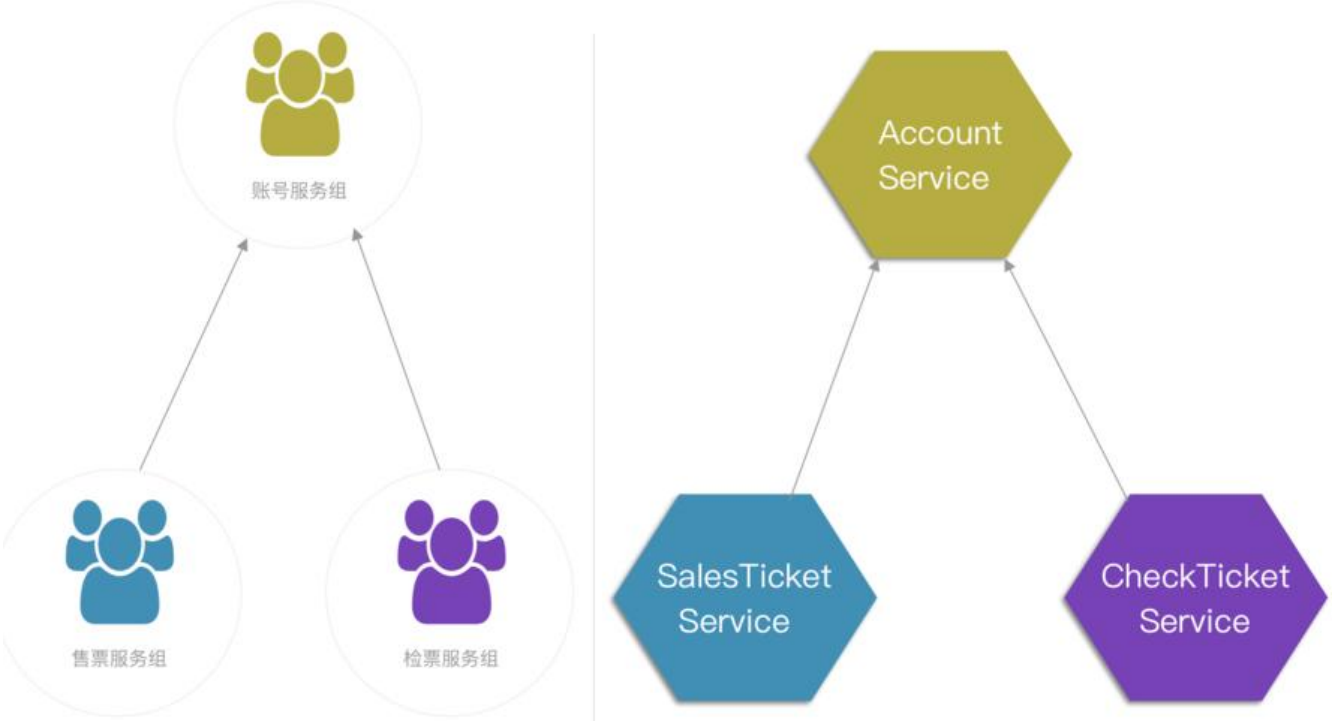
订单服务和数据访问服务，前者反映了真是的订单服务，后者则是一种技术抽象服务，不能反映真是业务，所有按微服务架构的理念来划分时，是不应该存在数据访问服务这样的一个服务的。

事实上，传统应用设计架构的分层结构，正反映了不同角色的沟通结构，所有若按微服务的方式来构应用，也需要对应调整团队的组织架构，每个服务背后的小团队的组织是跨功能的，包含实现业务所的全面技能。

传统组织结构

传统应用架构的组织架构，按照职能划分，如UI组，开发组，DBA组，运维组等等，

微服务组织结构



微服务架构模式下的组织架构下，每个服务背后的小团队的组织是跨功能的，包含实现业务所需的全技能。