

Microservice

著作权归作者所有。

商业转载请联系作者获得授权，非商业转载请注明出处。

作者：用心阁

链接：<http://www.zhihu.com/question/37808426/answer/81212783>

来源：知乎

SOA

SOA的提出是在企业计算领域，就是要将紧耦合的系统，划分为面向业务的，粗粒度，松耦合，无状态的服务。服务发布出来供其他服务调用，一组互相依赖的服务就构成了SOA架构下的系统。

基于这些基础的服务，可以将业务过程用类似BPEL流程的方式编排起来，而BPEL反映的是业务处理的过程，这些过程对于业务人员更为直观，调整也比hardcode的代码更容易。

当然企业还需要对服务治理，比如服务注册库，监控管理等。

我们知道企业计算领域，如果不是交易系统的话，并发量都不是很大的，所以大多数情况下，一台服务器就容纳将许许多多的服务，这些服务采用统一的基础设施，可能都运行在一个应用服务器的进程中。虽然说是面向服务了，但还是单一的系统。

微服务

而微服务架构大体是从互联网企业兴起的，由于大规模用户，对分布式系统的要求很高，如果像企业计算那样的系统，伸缩就需要多个容纳越来越多的服务的系统实例，前面通过负载均衡使得多个系统成为一个集群。

但这是很不方便的，互联网企业迭代的周期很短，一周可能发布一个版本，甚至可能每天一个版本，而不同的子系统的发布周期是不一样的。

而且，不同的子系统也不像原来企业计算那样采用集中式的存储，使用昂贵的Oracle存储整个系统的数据，二是使用MongoDB，HBase，Cassandra等NOSQL数据库和Redis，memcache等分布式缓存。

那么就倾向采用以子系统为分割，不同的子系统采用自己的架构，那么各个服务运行在自己的Web容器中，当需要增加计算能力的时候，只需要增加这个子系统或服务的实例就好了，当升级的时候，可以不影响别的子系统。这种组织方式大体上就被称作微服务架构。

微服务与SOA相比，更强调分布式系统的特性，比如横向伸缩性，服务发现，负载均衡，故障转移，高可用。互联网开发对服务治理提出了更多的要求，比如多版本，比如灰度升级，比如服务降级，比如分布式跟踪，这些都是在SOA实践中重视不够的。

Docker容器技术的出现，为微服务提供了更便利的条件，比如更小的部署单元，每个服务可以通过类似Node.js或Spring Boot的技术跑在自己的进程中。可能在几十台计算机中运行成千上万个Docker容器，每个容器都运行着服务的一个实例。随时可以增加某个服务的实例数，或者某个实例崩溃后，在其他的计算机上再创建该服务的新的实例。

这就是我对SOA和微服务架构区别的一点理解。

Componentization via Services

Organized around Business Capabilities

Products not Projects

Smart endpoints and dumb pipes

Decentralized Governance

Decentralized Data Management

Infrastructure Automation

Design for failure

Evolutionary Design

