

84 | 程序员练级攻略：微服务

2018-7-19 陈皓

微服务是分布式系统中最近比较流行的架构模型，也是SOA架构的一个进化。微服务架构并不是银弹，所以，也不要寄希望于微服务架构能够解决所有的问题。微服务架构主要解决的是如何快速地开发和部署我们的服务，这对于一个能够适应快速开发和成长的公司是非常必要的。同时我也觉得，微服务中有很多很不错的想法和理念，所以学习微服务是每一个技术人员迈向卓越的架构师的必经之路。

首先，你需要看一下，Martin Fowler的这篇关于微服务架构的文档 - [Microservice Architecture](#)（[中译版](#)），这篇文章说明了微服务的架构与传统架构的不同之处在于，微服务的每个服务与其数据库都是独立的，可以无依赖地进行部署。你也可以看看Martin Fowler老人家现身说法的[视频](#)。

另外，你还可以简单地浏览一下，各家对微服务的理解。

[AWS的理解 - What are Microservices?](#)。

[Microsoft的理解 - Microservices architecture style](#)。

[Pivotal的理解 - Microservices](#)。

微服务架构

接下来，你可以看一下 [IBM红皮书：Microservices Best Practices for Java](#)，这本书非常好，不但有通过把Spring Boot和 Dropwizard来架建Java的微服务，而且还谈到了一些标准的架构模型，如服务注册、服务发现、API网关、服务通讯、数据处理、应用安全、测试、部署、运维等，是相当不错的一本书。

当然，有一本书你也可以读一下——[微服务设计](#)。这本书全面介绍了微服务的建模、集成、测试、部署和监控，通过一个虚构的公司讲解了如何建立微服务架构。主要内容包括认识微服务在保证系统设计与组织目标统一上的重要性，学会把服务集成到已有系统中，采用递增手段拆分单块大型应用，通过持续集成部署微服务，等等。

与此相似的，也有其它的一系列文章，值得一读。

下面是Nginx上的一组微服务架构的系列文章。

[Introduction to Microservices](#)

[Building Microservices: Using an API Gateway](#)

[Building Microservices: Inter-Process Communication in a Microservices Architecture](#)

[Service Discovery in a Microservices Architecture](#)

[Event-Driven Data Management for Microservices](#)

[Choosing a Microservices Deployment Strategy](#)

[Refactoring a Monolith into Microservices](#)

下面这是 [Auto0 Blog](#) 上一系列的微服务的介绍，有代码演示。

[An Introduction to Microservices, Part 1](#)

[API Gateway. An Introduction to Microservices, Part 2](#)

[An Introduction to Microservices, Part 3: The Service Registry](#)

[Intro to Microservices, Part 4: Dependencies and Data Sharing](#)

[API Gateway: the Microservices Superglue](#)

还有Dzone的这个Spring Boot的教程。

[Microservices With Spring Boot - Part 1 - Getting Started](#)

[Microservices With Spring Boot - Part 2 - Creating a Forex Microservice](#)

[Microservices With Spring Boot - Part 3 - Creating Currency Conversion Microservice](#)

[Microservices With Spring Boot - Part 4 - Using Ribbon for Load Balancing](#)

[Microservices With Spring Boot - Part 5 - Using Eureka Naming Server](#)

当然，如果你要玩得时髦一些的话，我推荐你使用下面的这套架构。

前端：[React.js](#) 或 [Vue.js](#)。

后端：[Go语言](#) + 微服务工具集 [Go kit](#)，因为是微服务了，所以，每个服务的代码就简单了。既然简单了，也就可以用任何语言了，所以，我推荐Go语言。

通讯：[gRPC](#)，这是Google远程调用的一个框架，它比Restful的调用要快20倍到50倍的样子。

API：[Swagger](#)，Swagger是一种Restful API的简单但强大的表示方式，标准的，语言无关，这种表示方式不但人可读，而且机器可读。可以作为Restful API的交互式文档，也可以作为Restful API形式化的接口描述，生成客户端和服务端的代码。今天，所有的API应该都通过Swagger来完成。

网关：[Envoy](#) 其包含了服务发现、负载均衡和熔断等这些特性，也是一个很有潜力的网关。当然，Kubernetes也是很好的，而且它也是高扩展的，所以，完全可以把Envoy通过Ingress集成进Kubernetes。这里有一个开源项目就是干这个事的 - [contour](#)。

日志监控：[fluentd](#) + [ELK](#)。

指标监控：[Prometheus](#)。

调用跟踪：[Jaeger](#) 或是 [Zipkin](#)，当然，后者比较传统一些，前者比较时髦，最重要的是，其可以和Prometheus和Envoy集成。

自动化运维：[Docker](#) + [Kubernetes](#)。

微服务和SOA

在对微服务有了一定的认识以后，一定有很多同学分不清楚微服务和SOA架构，对此，你可以看一下这本电子书 - 《[Microservices vs. Service-Oriented Architecture](#)》。通过这本书，你可以学到，服务化架构的一些事实，还有基础的SOA和微服务的架构知识，以及两种架构的不同。这本书的作者马克·理查兹（Mark Richards）同学拥有十年以上的SOA和微服务架构的设计和实现的经验。

另外，还有几篇其它对比SOA和微服务的文章你也可以看看。

[DZone: Microservices vs. SOA](#)

[DZone: Microservices vs. SOA - Is There Any Difference at All?](#)

[Microservices, SOA, and APIs: Friends or enemies?](#)

除此之外，我们还需要知道微服务和其它架构的一些不同和比较，这样我们就可以了解微服务架构的优缺点。下面几篇文章将帮助获得这些知识。

[PaaS vs. IaaS for Microservices Architectures: Top 6 Differences](#)

[Microservices vs. Monolithic Architectures: Pros, Cons, and How Cloud Foundry \(PaaS\) Can Help](#)

[Microservices - Not A Free Lunch!](#)

[The Hidden Costs Of Microservices](#)

设计模式和最佳实践

然后，你可以看一下微服务的一些设计模式。

[Microservice Patterns](#)，微服务架构的设计模式和最佳实践。

[Microservice Antipatterns and Pitfalls](#)，微服务架构的一些已知的反模式和陷阱。

[Microservice Architecture: All The Best Practices You Need To Know](#)，这是一篇长文，里面讲述了什么是微服务、微服务架构的优缺点、微服务最大的挑战和解决方案是什么、如何避免出错，以及构建微服务架构的最佳实践等多方面的内容。推荐阅读。

[Best Practices for Building a Microservice Architecture](#)，这篇文章分享了构建微服务架构的最佳实践。

[Simplicity by Distributing Complexity](#)，这是一篇讲如何使用事件驱动构建微服务架构的文章，其中有很多不错的设计上的基本原则。

相关资源

[Microservices Resource Guide](#)，这个网页上是Martin Fowler为我们挑选的和微服务相关的文章、视频、书或是podcast。

[Awesome Microservices](#)，一个各种微服务资源和相关项目的集中地。

小结

好了，总结一下今天的内容。我认为，微服务中有很多很不错的想法和理念，所以学习微服务是每一个技术人员迈向卓越的架构师的必经之路。在这篇文章中，我先给出了AWS、Microsoft和Pivotal对微服务的理解；然后给出了好几个系列的教程，帮你全面学习和理解微服务架构；然后通过一系列文章帮你来区分何为微服务，何为SOA；最后给出了微服务架构的设计模式和最佳实践，以及相关资源。相信通过这一系列内容的学习，你一定会对微服务有全面、透彻的理解。

下篇文章，我们将讲述的容器化和自动化运维方面的内容。敬请期待。

下面是《程序员练级攻略》系列文章的目录。

[开篇词](#)

入门篇

[零基础启蒙](#)

[正式入门](#)

修养篇

[程序员修养](#)

专业基础篇

[编程语言](#)

[理论学科](#)

[系统知识](#)

软件设计篇

[软件设计](#)

高手成长篇

[Linux系统、内存和网络（系统底层知识）](#)

[异步I/O模型和Lock-Free编程（系统底层知识）](#)

[Java底层知识](#)

[数据库](#)

[分布式架构入门（分布式架构）](#)

[分布式架构经典图书和论文（分布式架构）](#)

[分布式架构工程设计\(分布式架构\)](#)

[微服务](#)

[容器化和自动化运维](#)

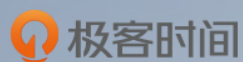
[机器学习和人工智能](#)

[前端基础和底层原理（前端方向）](#)

[前端性能优化和框架（前端方向）](#)

[UI/UX设计（前端方向）](#)

[技术资源集散地](#)



左耳朵耗子

全年独家专栏《左耳听风》

20000 名程序员的练级攻略

陈皓

资深技术专家
骨灰级程序员



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。

精选留言 21



Yangjing

1531976164

用Go的点赞，看有多少人了



离空气

1532306137

越来越觉得没什么干货了，都是罗列一些网站放在上面，没必要，我们要看的并不是这些





Michael
1531994915

30多的大龄程序员，如果能把耗哥前面提到的 专业基础部分都掌握透了，还有前途嘛
现在很多招聘对年龄都有要求，感觉前途有点担忧



小薛薛
1531960792

老师什么时候能讲讲你的学习方法和时间管理吗？ 因为我觉得学习效率很重要呢！



bensinos
1531965221

感觉Go会越来越流行



天地乾坤
1531958745

太有帮助了，微服务是组成业务流程的最小自治域。是对传统mvc模式的颠覆。新项目，新架构都应该着手拥抱它。



沫沫（美丽人生）
1533515382

陈老师，作为一个落地方案，一个小公司，使用spring cloud怎么样？

作者回复 可以啊



Hunter_wyg
1533002105

我觉得这些资料很有用，适合给对微服务有一些实践经验的工程师。相当于浩哥整理了知识框架，筛选出了有效资源，我们花点时间精读一下就能点亮知识地图。



小鱼儿
1532613636

资料太多，可以精炼些！更多希望是老师自己经验和理解！

作者回复 这已经删掉了7成了



手打丸子

1542160365

后几篇初看觉得是凑数的，迷茫时再看，简直指了条明路。不仅指明了方向，还上好了菜放好了资源。从业五年的飘过。



hehai

1536071929

搜索一下microservices，前两页就是这些文章。希望看到干货



yanfan_software

1532044934

微服务 配合 Docker 感觉已经是趋势了。。其实应该提提 Spring Cloud，主流组合



godtrue

1547165916

阅后留痕

文字很简，内容巨多，



江龙

1535989224

有个api的设计原则问题，请教下，就是app的某些页面其实有多个资源聚合，那是应该app端去请求多个资源服务，然后聚合出来展示；还是有个后台服务去聚合后台的各个基础服务，然后只提供一个接口给app访问？这其中的粒度如何把握？



escray

1535672758

正在犹豫是否要买极客时间上的微服务专栏，价格有优惠，内容看上去也还好，只是自己并没有这方面的工作机会。回顾了一下这篇文章，感觉不用买了，先把这些来自源头的干货看了再说



豹豹

1534340354

ts + node 做微服务用哪些哈？



sipom

1532391425

真的建议浩哥增加分享一些在自己工作中运用这些技术的心得体会、经验教训，这些信息对缺少经验的人很难得、很珍贵。

作者回复 对于分布式微服的坑和经验教训，注意点，难题，解决方案，我全部写在了之前的分布式系列的文章中了。



xpisme

1532363911

工作两三年的，跟上有点费力。
英语是进阶高级工作师，架构师的毕竟之路。



超杰

1532316368

现在正在学习皓哥推荐的书籍，有读书群可以加入嘛



刘杰

1532253165

《微服务设计》的中文版翻译很烂，很业余，各位尽量读英文版吧



superK

1531994064

react有vue-cli一样的脚手架吗？找了半天没找到