

46 | 服务端开发篇：回顾与总结

2019-10-01 许式伟

许式伟的架构课

[进入课程 >](#)



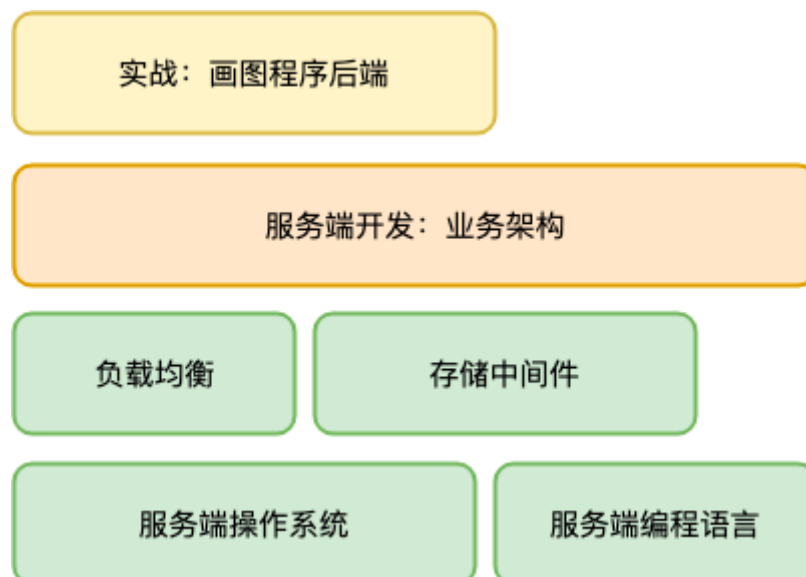
讲述：姚迪迈

时长 08:44 大小 8.02M



你好，我是七牛云许式伟。

到今天为止，我们第三章“服务端开发篇”就要结束了。今天，让我们对整章的内容做一个回顾与总结。本章我们主要涉及的内容如下。



服务端开发这个分工，出现的历史极短。如果我们从互联网诞生算起也就 40 多年的历史。以进入民用市场为标志，它真正活跃的时段，其实只有 20 多年。

作为架构师，记住这一点非常非常重要。20 多年能够形成的有效经验并不多。这意味着我们不能固步自封，很多惯例是可以被挑战的，并且最终也必然被挑战的。

作为最底层的服务端操作系统，最初从桌面操作系统而来。但桌面操作系统自身在发展，服务端操作系统自身也在发展，两者渐行渐远。

桌面的领域特征是强交互，以事件为输入，GDI 为输出。

所以，桌面技术的迭代，是交互的迭代，是人机交互的革命。在“[13 | 进程间的同步互斥、资源共享与通讯](#)”一讲中，我们介绍了桌面操作系统中进程间协同方式的变迁。如果我们从业务需求角度看，这个变迁本质上也是交互的变迁（为什么我们这么说？欢迎留言探讨）。

而服务端程序有很强烈的服务特征。它的领域特征是大规模的用户请求，以及 24 小时不间断的服务。这些都不是业务功能上的需要，是客户服务的需要。

所以，服务端技术的迭代，虽然一开始沿用了桌面操作系统的整套体系框架，但它正逐步和桌面操作系统分道而行，转向数据中心操作系统（DCOS）之路。

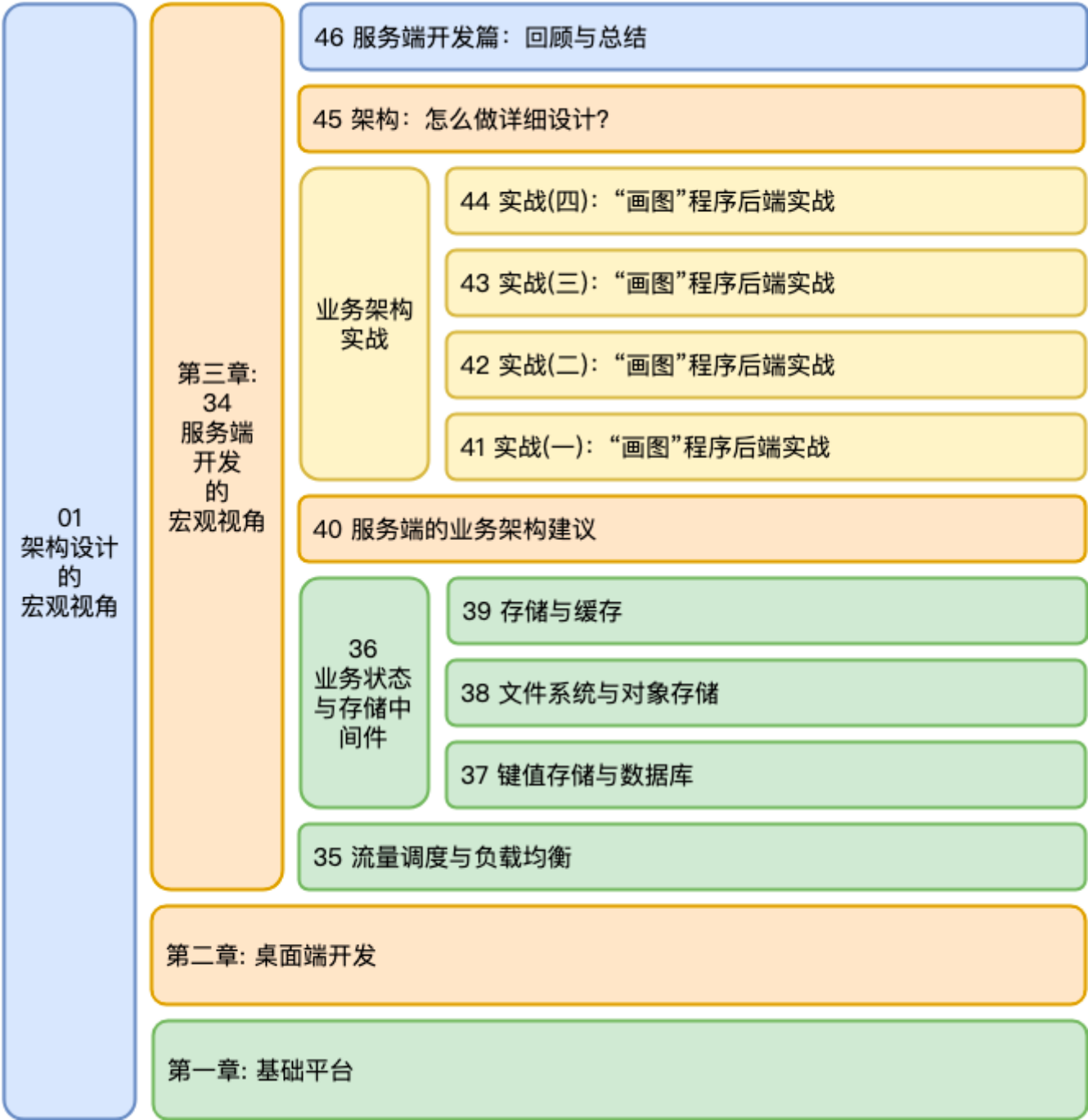
服务端技术的迭代，有一些和服务端开发相关，会影响到业务架构。而更多则和业务架构无关，属于服务治理的范畴。

服务端开发与服务治理的边界在于，服务端开发致力于设计合适的业务架构来满足用户需求，而服务治理则致力于让服务端程序健康地为客户提供不间断的服务。

关于服务治理相关的内容，我们留到下一章来介绍。

服务端开发篇的内容回顾

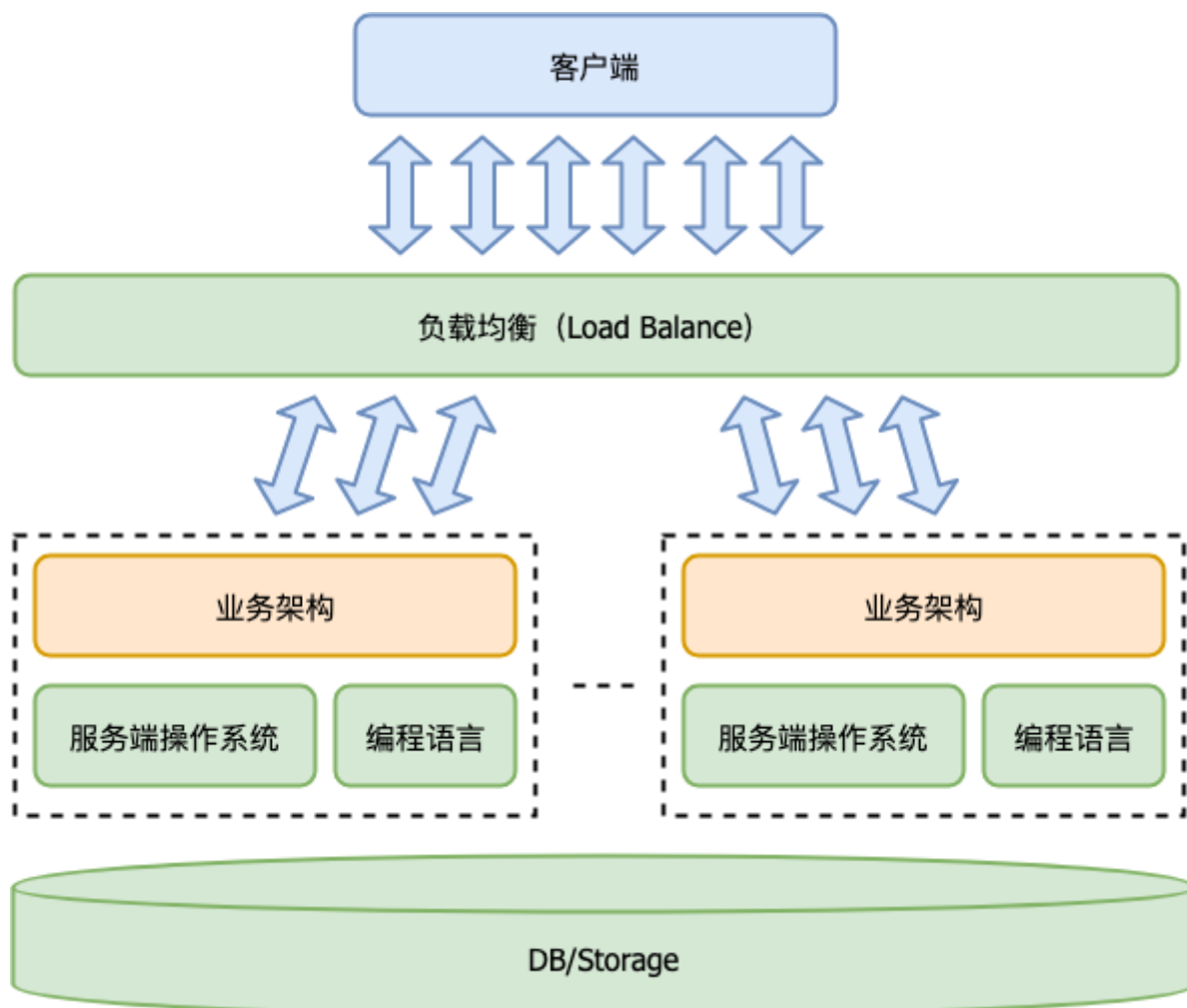
本章服务端开发篇我们讲了些什么？为了让你对第三章内容有个宏观的了解，我画了一幅图，如下。



首先，从服务端开发来说，服务端程序依赖的基础软件不只是操作系统和编程语言，还多了两类：

负载均衡（Load Balance）；

存储中间件：数据库或其他形式的存储（DB/Storage）。



负载均衡的最大价值是对客户的访问流量进行调度，让多个业务服务器的压力均衡。这里面隐含的一个前提是负载均衡软件的抗压能力往往比业务服务器强很多。这表现在：

其一，负载均衡的实例数 / 业务服务器的实例数往往大大小于 1；其二，DNS 的调度不均衡，所以负载均衡的不同实例的压力不均衡，有的实例可能压力很大。

当然，负载均衡的价值并不只是做流量的均衡调度，它也让我们的业务服务器优雅升级成为可能。

存储中间件即数据结构。

在服务端开发领域，有一个很知名的编程哲学，叫“速错（Fail Fast）”，它的核心逻辑是，一旦发生非预期的错误时，应该立刻退出程序，而不要尝试为该错误去写防御代码，因为那样的话掩盖掉这个错误，并导致后续可能产生更隐晦难以定位的错误。

但是“速错（Fail Fast）”是以可靠的存储中间件为前提的。没有了可靠的存储，程序重新启动后就不知道自己正在做什么事情了。所以存储是不能速错的，它的编程哲学如此不同。作为存储系统的开发者，你需要花费绝大部分精力在各种异常情况的处理上，甚至你应该认为，这些庞杂的、多样的错误分支处理，才是存储系统的“正常业务逻辑”。

对于服务端来说，存储中间件至关重要，它是服务端程序能够提供高并发访问和 24 小时不间断服务的基础。存储中间件极大地解放了生产效率，让开发人员可以把精力放在具体的业务需求上。

虽然我们不需要自己去开发存储中间件，但是深度理解其工作原理是非常有必要的。通常来说，存储中间件也是服务端的性能瓶颈所在。几乎所有服务端程序扛不住压力，往往都是因为存储没有扛住压力。

存储中间件种类繁多，不完整的列表如下：

键值存储（KV-Storage）；

对象存储（Object Storage）；

数据库（Database）；

消息队列（MQ）；

倒排索引（SearchEngine）；

...

对象存储的出现，是服务端体系架构和桌面操作系统分道扬镳的开始。文件系统（File System）不再是服务端存储中间件的标配。第一个大家公认的对象存储是 AWS S3，但它只是一个基础文件存取组件。七牛云则在此基础上推出了第一个“对象存储 + CDN + 多媒体处理”融合的 PaaS 型云存储。

理解了负载均衡和存储中间件，我们开始谈[服务端的业务架构](#)。

从业务架构的角度，服务端主要是实现一个多租户的 Model 层。Model 层本身最重要的是自然体现业务逻辑，它和具体行业的领域问题相关。但服务端程序还是有它很鲜明的特点，有一些和领域无关的业务架构通用问题。比如：网络协议、帐号与授权、RPC 框架、单元测试等等。

为了更好地理解服务端开发的架构逻辑，我们继续以画图程序的后端开发为实战案例，进行详细展开。

作为最后收官，我们聊了架构[第三步：详细设计](#)。详细设计关注的是子系统或模块的全貌。它并不是只谈实现就完事，更不是一个架构图。它包括以下这些内容。

现状与需求

现在在哪里，遇到了什么问题，要作何改进。

需求满足方式

要做成啥样？交付物的规格，或者说使用界面（接口）。

怎么做到？交付物的实现原理。

“程序 = 数据结构 + 算法” 是我们很熟悉的一个公式。它其实是怎么描述实现原理的很好的指导方针。当我们谈程序的实现时，我们总是从数据结构和算法两个维度去描述它。

服务端开发篇的参考资料

整体来说，尽管服务端开发所需要的知识面更广，但是就开发本身的工作量和难度而言，服务端开发要大大低于桌面开发。

但将服务端程序开发出来只是个开始。如何让服务稳定健康地运行，是一个复杂的话题。所以近年来服务端技术蓬勃发展，主要以服务治理为主。

单单从服务端开发的角度，我们除了关注服务端操作系统、编程语言，还需要关注负载均衡和存储中间件。

这里我列一下我认为值得重点关注的技术：

Docker & Kubernetes。毫无疑问，数据中心操作系统（DCOS）是服务端操作系统的发展方向。关于 DCOS，我们会在下一章涉及。

Go 语言。推荐 Brian W. Kernighan 写的《Go 程序设计语言》，本书为传世经典《C 程序设计语言》的作者再次动笔所创。

LVS & Nginx。两大当前最主流的流量调度软件。其中 LVS 工作在网络层，Nginx 工作在应用层。

MySQL & MongoDB。两大当前最主流的数据库。虽然它们的使用范式差异较大，但背后的基础哲学实际上是相通的。

对象存储。推荐 AWS S3 和 [七牛云存储](#)。

网络协议。虽然当前主流还是 RESTful API，但可以适当关注 [GraphQL](#)。

RPC 框架。推荐七牛云开源的 [restrpc](#)，以及 Google 开源的 [grpc](#)。

HTTP 测试。推荐七牛云开源的 [httptest](#) 框架和 [qiniutest](#) 实用程序。

大部分的服务端技术都还在快速迭代。对于网络资料相对较多的部分，这里我就不再给出具体的相关资料了。

结语

今天我们对本章内容做了概要的回顾，并借此对整个服务端开发的骨架进行了一次梳理。

这一章我们继续聊业务架构，我们把侧重点放在后端业务开发。学业务架构最好的方式是：“做中学”。做是最重要的，然后要有做后的反思，去思考并完善自己的理论体系。

如果你对今天的内容有什么思考与解读，欢迎给我留言，我们一起讨论。下一讲我们开始进入第四章：服务治理篇。


如果你觉得有所收获，也欢迎把文章分享给你的朋友。感谢你的收听，我们下期再见。

许式伟的架构课

从源头出发, 带你重新理解架构设计

许式伟
七牛云 CEO



新版升级: 点击「 请朋友读」, 20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 45 | 架构: 怎么做详细设计?

下一篇 加餐 | 如何做HTTP服务的测试?

精选留言 (5)

写留言



Louis

2019-10-03

问下许老师, 服务端治理是不是就是运维?

展开

作者回复: 服务治理包括运维



1



随心而至

2019-10-02

我刚工作一年多, 做的服务端开发, 刚开始技术繁杂, 令人眼花缭乱, 学习老师的专栏, 暂时就是知道该学习那些知识, 什么个次序, 至于架构师应该是水到渠成的事情。



Aaron Cheung

2019-10-01

不能固步自封 挑战部分惯例 国庆快乐 打卡46

展开 ∨



Geek_88604f

2019-10-04

桌面技术的迭代是交互的迭代——为什么这么说？结合老师以前的文章，我说一下个人的理解，桌面技术可以划分为桌面应用和桌面交互两个方面，桌面技术的发现和演进必然是由这两个方面共同驱动的，它俩像人的两条腿一样共同带领桌面技术稳步前进。

先来看桌面应用，如果按照网络发展的时间轴来划分桌面应用，大致可以划分为单机应用、网络应用、移动应用这三个时期。在单机应用时代桌面应用有文本编辑、语音、...

展开 ∨



小名叫大明

2019-10-02

老师，很感谢您推荐的值得重点关注的技术列表，我深知他们的重要性，可是我有一个疑惑，了解到什么程度呢？

是越深越好哈？还是应用级别，浅层原理级别，深层原理级别？

展开 ∨

作者回复: 以串联知识为要，深入的那部分，应该是对你业务最相关或者是你兴趣最相关。

