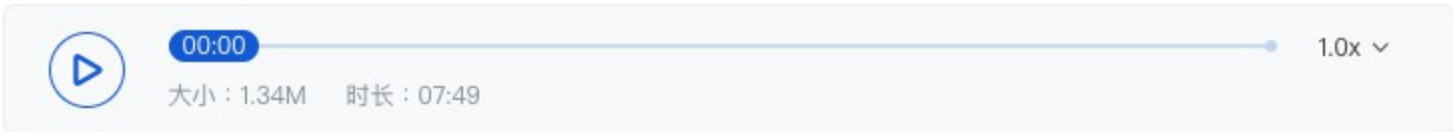




## 亚马逊 CTO 20 年架构经验之道：俭约架构师的七大黄金法则！

作者：Tina

2023-12-01 · 北京 · 本文字数：2684 字 · 阅读完需：约 9 分钟



亚马逊 CTO Werner Vogels 向企业传达了一条信息：在管理云成本方面，是时候成为节俭的架构师了。

他拥有近 20 年的平台构建经验，在今天的 re:Invent 2023 大会主题演讲中，给大家上了一节关于成本优化的课：“作为技术专家，我们生活在一个瞬息万变的世界，我们需要保持学习，坐下来，拿出你的记事本，现在开始做笔记。”



Vogels 选择在主题演讲中讨论成本问题，这既反映了当前的经济环境，也反映了云计算支出的不断增长的态势。本月早些时候，Gartner 公司发布预测，到 2024 年云用户支出将达到 6780 亿美元，比今年的 5630 亿美元大幅增长。

亚马逊云科技在引领公有云市场方面取得了巨大成功，但同时也意识到，这个行业所带来的成本压力正在随着生成式 AI 等技术的广泛采用而不断增加。

“在构建这些系统、在有限的环境中生活中蕴含了很多艺术，”Vogels 表示。“云计算消除了所有这些限制。突然间，最重要的事情就是迅速行动，推出产品。随着执行速度变得更为重要，我们失去了关注成本、以成本为重要考量的架构设计的艺术。作为构建者，我们确实需要开始思考这一点。”

Vogels 概述了成为“俭约架构师”（The Frugal Architect）的七条关键法则，这是一组他自己描述的“法则”，已发布在一个专门致力于此主题的新网站上。其中包括创建将成本与业务对齐的系统，观察基础设施中的关键运营网络以避免未知的费用，并追求渐进式优化。

“我尝试向初创公司强调这一点，”Vogels 说。“你将要采用什么收入模式？他们需要构建符合这一模式的架构。确保你获取收入的维度始终与你的成本保持一致。”

附完整的“俭约架构师的七大黄金法则”：

### 推荐阅读

打造一个成本优先的技术架构，可以分几步？

🔗 架构，云原生

34 | 落地经验：怎样在实际项目中推广 DDD？

2023-03-02

09. 读懂财务报表会获得哪些收益？

2023-10-17

结束语 | 以重构为动力，让遗留系统焕发新生

2023-04-12

老架构师总结的 12 个软件架构陷阱 | 避坑指南

🔗 业务架构

在架构方面，没有谁比亚马逊的 CTO 更能预见未来

🔗 架构

一行代码价值百万美元：从工程技术角度看云成本优化

🔗 AI&大模型，后端

### 电子书



#### 中国开发者画像洞察研究报告 2024

分析开发者的行为模式、工作价值、职业发展等内容，帮助整个行业生态更深入地理解开发者，为他们提供更精准...

立即下载

### 大厂实战PPT下载

换一换



#### 传统业务系统奔向云原生的“纵”“横”架构实践

裴斐 | 网易 高级技术专家，资深架构师

立即下载

#### 中国移动智慧中台赋能企业数智化转型实践

兰建明 | 中国移动 信息技术中心/架构师

立即下载

#### 21天 AIGC 行动营

陈旻 | 清华大学计算机博士、阿里云MVP  
《21天 AIGC 行动营》专栏作者

立即下载





架构师如何培养自己的成本意识、增强现代架构可持续性。

## 法则一

将成本视为一种非功能性需求。

所谓非功能性需求，就是用于判断系统操作的标准，与具体特性或功能无关。可访问性、可用性、可扩展性、安全性、可移植性、可维护性和合规性等都在此列。而成本往往是其中受到忽略的一条。

业务之所以身陷困境，往往是因为他们没有考虑到各个阶段中的相应成本——从设计到开发、再到运营——也可能是未能正确量化成本。这背后的原理非常简单：如果成本比收入还高，那还做业务干嘛呢。更早、以更加可持续的方式考虑成本影响，架构师才能在系统设计过程中在功能、上市时间和效率之间寻求平衡。这样开发团队可以专注于维护更加精简高效的代码，运营部门可以优化资源用量和支出，从而最大限度提高盈利能力。

## 法则二

确保系统的最终成本与业务保持一致。

系统能否长治久安，取决于其成本是否与业务模式高度匹配。在设计和构建系统时，架构师必须考虑收入来源和利润杠杆。更重要的是，必须找到能够产生利润的维度，确保架构规划始终围绕收益展开。

例如，在电子商务领域，这个核心维度可能是订单数量。当订单增加时，基础设施和运营成本也会随之上升。但没关系，只要系统架构设计良好，我们就能享受规模经济带来的红利。最重要的是基础设施成本对业务的影响始终精确、可以量化。

作为架构师，大家需要关注收入，并据此指导技术选型。任何不计代价的增长只会招致毁灭。

## 法则三

架构设计是一系列权衡的集合。

在架构当中，每项决定都涉及相应的权衡。成本、弹性和性能这些非功能性需求之间，往往相互冲突、难以调和。

常言道“万事万物终将陨落。”要想抵御这种失败的风险，就必须关注弹性，同时牺牲掉一部分性能。

在技术与业务需求间找到适当的平衡将至关重要，也就是把握住风险承受能力与预算额度间的最佳比例。请记住，节俭是为了最大限度提升价值，而不只是尽可能控制支出。因此，在必须得花的钱上别吝啬。

## 法则四

无法观察的系统将带来无法估量的成本。

如果不认真观察和测量，系统运营的真实成本将难以把控。就如同隐藏在地下室中的电表一样，这种直观性的缺失必然导致浪费。所以一定要把指标摆在明面上，这将深刻改变运营行为。

尽管实现可观察性需要投入，但这笔钱绝对会物有所值。有句格言说“如果无法量化，也就无法管理。”请始终坚持对利用率、支出、错误等至关重要的成本管理指标保持关注。

当工程师和业务合作伙伴能够随时查看关键成本指标时，自然就会催生出更具可持续性的实践策略。持续检查能帮我们发现非必要支出，并调整运营以减少浪费。总之，可观察性带来的回报往往远超过前期投入。

最重要的是，这本身也是对成本的强调，能在企业中塑造出鼓励可持续性实践的文化。

## 法则五

依托成本感知架构实现成本控制。





节俭架构的本质，在于强大监控与成本优化能力的结合。精心设计的系统能帮助大家抓住改进的机会。为此，请将应用程序拆分成一系列可以调节的构建块。

一种常见的方法就是按重要性对组件进行分层。T1 层组件必不可少，应当不计成本进行优化；T2 层组件非常重要，但暂时缩小规模也不会产生重大影响；T3 层组件则属于“锦上添花”，要保证其成本低廉且易于控制。

明确定义各层，即可在成本及其他要求之间求得平衡。对组件的精细控制则能优化成本和体验。基础设施、语言、数据库都应具备可调节性，并在系统的设计和构建阶段考虑收入和利润。总之，成本优化必须可量化，且与业务影响直接挂钩。

## 法则六

成本优化是个渐进的过程。

追求成本效率是个持续的过程。即使在部署之后，我们也必须随时审视系统以逐步寻求优化。其中的关键在于不断提问、深入研究。编程语言往往提供分析工具以追踪代码性能，虽然这需要额外的精力和专业知识，但精细的调优足以带来几毫秒的差异。而这种看似微小的优化，累积起来足以产生超出想象的成本优势。

在运营中，大部分时间都被用于运行现有系统。所以请把握一切机会，分析资源使用情况并减少浪费。在亚马逊，我们持续监控生产中的服务，发现运营模式并消除低效因素。节俭是坚持的结果——通过逐步降低延迟和基础设施成本，服务成本才能最终得到优化。

只要不懈努力，我们总能找到改进空间。而今天省下的资源，就是明天创新的燃料。

## 法则七

顺风局打多了会让人盲目自信。

如果软件团队在取得重大成功的过程中，从未经历过任何严重失败或者阻碍，则往往会出现自满情绪。这是一种危险的倾向，会导致团队成员对原本的方法、工具和实践变得盲目信息。

软件团队经常陷入这样的陷阱：仅凭以往的工作经验，他们就认为当前的技术、架构或语言永远是最佳选择。这可能会产生一种错误的安全感，阻碍对现状的质疑，更会打击对可能更加高效、更具成本效益或可扩展性更强的新选项的探索。

说到编程语言，人们往往会说“我们是一家 Java 公司”。这话大有问题，其底层逻辑无疑是在扼杀创新。唾手可得的成功会滋生自满情绪，而只有质疑才能不断激发新的优化与改进思路。

Grace Hopper 的名言，准确反映了这一值得高度警惕的陷阱：“我们一直都是这样做的。”

发布于：2023-12-01 09:38 | 阅读数：14199  
文章版权归极客邦科技InfoQ所有，未经许可不得转载。

🔗 框架 服务革新 架构 云计算

👍 轻点一下，留下你的鼓励

### 评论 1 条评论

写下你的想法，一起交流

发布



轻舞凋零

面对当前下云的说法。是想说：上云成本不高，其实是你的架构存在问题吗？

2023-12-01 21:20 · 北京

👍 4 回复

没有更多了

### 更多内容推荐

#### 写给架构师的技术债“偿还”指南

聊聊技术债务的定义，影响以及管理。

🔗 架构, 文化 & 方法, 研发效能, 操作系统, 框架

#### 构建可观测平台：应该购买商业产品还是自己搭建？

建立可观测性，是应该自己搭建，还是直接购买商业产品呢？这节课，我们就来聊聊这个问题。

2022-09-23

#### 数据中台在网易电商业务的最佳实践

今天这节课，我会从项目立项、到项目推进，最后到项目总结，带你看一下网易电商数据中台的构建过程。

2020-05-06

#### 成本：马斯克的猎鹰火箭为什么可以这么便宜？

马斯克的技术并不占行业优势，而是通过很多低技术的组合实现了对高技术能力的超越，这是一个商业思路大胆的...

2022-07-18

#### 质量与速度的均衡：让“唯快不破”快得更持久

在恰当的时间“举债前行”，而在平时的开发工作中要持续定位技术债任务，并解决高优先级的部分，才能让开发快...

2019-09-23



架构师角色的演变：从发号施令到与团队合作

本文探讨了共享架构的文化和架构师角色的演变。

🔗 架构, 文化 & 方法, 性能优化, 编程语言, 框架, 微服务, 芯片&算力

不要让框架影响你最初的架构设计

在真正需要之前，不要对任何特定的框架、模式或策略过多投入。

🔗 架构, 文化 & 方法, 方法论, 最佳实践, 性能优化, 编程语言, 框架, 亚马逊云科技, 银行, 保险, 企业动态, 行业深度

打造高效对话，与商业利益相关者探讨软件架构

与商业利益相关者的任何沟通最终都会转换为与钱相关的对话。他们最关心的不是绝对成本，而是成本效益。

🔗 架构, 方法论, 性能优化, 框架, 芯片&算力, 银行, 保险, 汽车, 医疗, 企业动态

从微服务转为单体架构、成本降低 90%，亚马逊内部案例引发轰动！CTO：莫慌，要持...

人们之所以不公开谈论撤离微服务，是因为这可能会让一些人认为“他们做错了”。

🔗 云原生, 软件工程, 架构, 容器, 服务革新, 微服务, 最佳实践, 后端, 编程语言, 实时计算, 亚马逊云科技, 企业动态

千刀万剐的微服务，我们到底应该如何应对分布式系统的挑战和风险

软件行业再次认识到复杂性会导致失败。

🔗 微服务

平台工程的失败模式及如何避免，来自一线的宝贵经验

本文Aaron总结了导致平台工程失败的几种情况。

🔗 软件工程, DevOps & 平台工程, 最佳实践, 方法论, 框架, FinOps, 数字化转型, 团队搭建, 汽车, 企业动态, 行业...

时间刻度（下）：为什么优酷土豆会从行业第一下滑到行业第三？

时间的刻度可以说是一个衡量广告收入天花板的指标，不过，相比于广告模式，会员模式就可以解决一切问题了吗？

2022-08-15

上云设计，融合云计算的未来

专栏的最后一讲，我打算和你聊聊有关“上云设计”的话题。

2020-12-23

谷歌工作十年，我总结了这些工程师必备软技能

谷歌浏览器的一位高级工程经理，分享了他在谷歌工作十年时间里学到的一些软技能，下文为部分经验摘录。

🔗 语言 & 开发, 文化 & 方法, 架构, 技术管理, 软件工程, Chrome, 方法论, 最佳实践, 架构/框架, 性能优化, 编程...

软件架构决策指北：怀疑主义的软件架构设计

怀疑主义是一种架构超能力，可以帮助我们在错误的假设走得太远之前识破它们。

🔗 架构, 方法论, 性能优化, 框架, 团队搭建, 保险, 汽车

晟云磐盾郝峻晟：技术人如何走上创业之路，实现人生价值

InfoQ邀请了晟云磐盾总裁郝峻晟做了一场《技术人如何走上创业之路，实现人生价值》为题的分享。

🔗 多云/混合云, 管理/文化

只擅长构建软件是不够的，我们必须擅长构建可测试的软件 | QCon

你的测试问题实际上是可测性问题。

🔗 软件工程, QCon, 最佳实践, 可观测, 团队搭建, 企业动态, 行业深度

你的架构决策记录是否失去了它的目的？

架构决策涉及系统使用的基本概念，因为选择对代码的影响散布在整个软件中，而不是局部的。

🔗 业务架构

架构师（2023 年 12 月）

本期推荐文章：省钱在于“架构师”！亚马逊 CTO 20 年架构经验之道：俭约架构师的七大黄金法则；主力开发已...

🔗 业务架构, Serverless

“成本刺客”防不胜防，为何云成本如此难以驾驭？

云成本刺客防不胜防，如何能够在享受云计算这个巨大的创新引擎的同时又能够清晰度量云成本，不被“成本刺客”...

🔗 服务革新, 文化 & 方法, 技术管理, 最佳实践, 亚马逊云科技, 企业动态, 性能优化, 编程语言, 多云/混合云, 芯片...

发现更多内容



促进软件开发及相关领域知识与创新的传播

关于我们  
我要投稿  
合作伙伴  
加入我们  
关注我们

联系我们

内容投稿：editors@geekbang.com  
业务合作：hezuo@geekbang.com  
反馈投诉：feedback@geekbang.com  
加入我们：zhaopin@geekbang.com  
联系电话：010-64738142  
地址：北京市朝阳区望京北路9号2幢7层A701

InfoQ 近期会议

北京 · QCon 全球软件开发大会 2025.4.10-12  
上海 · AICon 全球人工智能开发与应用大会 2025.5.23-24  
北京 · AICon 全球人工智能开发与应用大会 2025.6.27-28

全球 InfoQ

🇨🇦 InfoQ En  
🇯🇵 InfoQ Jp  
🇫🇷 InfoQ Fr  
🇧🇷 InfoQ Br