

互联网创业核心技术

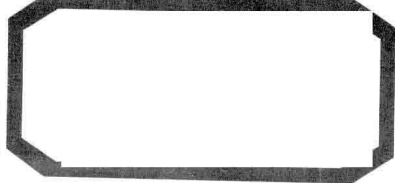
构建可伸缩的Web应用

【美】Artur Ejsmont 著 李智慧 何坤 译



Web Scalability for Startup Engineers
Tips & Techniques for Scaling Your Web Application

Web Scalability for
Tips & Techniques for Scaling



互联网创业核心技术

构建可伸缩的Web应用

【美】Artur Ejsmont 著 李智慧 何坤 译

电子工业出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

可伸缩架构技术是所有互联网技术中最重要,也是最引人入胜的技术。本书针对互联网创业需求快速迭代,业务快速发展,短时间内用户、数据、访问量激增的特点,提纲挈领地描述了伸缩性架构的基本原理与设计原则,详细阐述了Web应用前端层、服务层、数据层的可伸缩架构,并花大量篇幅讲述了缓存技术和异步处理技术的可伸缩设计及其在Web系统中的具体应用。

本书面向互联网创业公司工程师,也适用于所有互联网行业的工程师,对非互联网行业的软件工程师也有借鉴作用。事实上,本书适合所有对可伸缩架构有兴趣的软件技术人员阅读。

Artur Ejsmont, Web Scalability for Startup Engineers tips&Techniques for Scaling Your Web Application,ISBN 9780071843652,Copyright © 2015 by McGraw-Hill Education.

All Rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including without limitation photocopying, recording, taping, or any database, information or retrieval system, without the prior written permission of the publisher.

This authorized Chinese translation edition is jointly published by McGraw-Hill Education and Publishing House of Electronics Industry. This edition is authorized for sale in China Mainland.

Copyright © 2016 by McGraw-Hill Education and Publishing House of Electronics Industry.

版权所有。未经出版人事先书面许可,对本出版物的任何部分不得以任何方式或途径复制或传播,包括但不限于复印、录制、录音,或通过任何数据库、信息或可检索的系统。

本授权中文简体字翻译版由麦格劳-希尔(亚洲)教育出版公司和电子工业出版社合作出版。此版本经授权仅限在中国大陆销售。

版权©2016由麦格劳-希尔(亚洲)教育出版公司与电子工业出版社所有。

本书封面贴有McGraw-Hill Education公司防伪标签,无标签者不得销售。

版权贸易合同登记号 图字:01-2015-7884

图书在版编目(CIP)数据

互联网创业核心技术:构建可伸缩的web应用/(美)阿特·艾斯蒙特(Artur Ejsmont)著;李智慧,何坤译. —北京:电子工业出版社,2016.12

书名原文:Web Scalability for Startup Engineers

ISBN 978-7-121-30112-4

I. ①互… II. ①阿… ②李… ③何… III. ①网页制作工具—程序设计 IV. ①TP393.092.2

中国版本图书馆CIP数据核字(2016)第247615号

策划编辑:刘 蛟

责任编辑:郑柳洁

印 刷:北京季峰印刷有限公司

装 订:北京季峰印刷有限公司

出版发行:电子工业出版社

北京市海淀区万寿路173信箱 邮编:100036

开 本:720×1000 1/16 印张:21 字数:370.3千字

版 次:2016年12月第1版

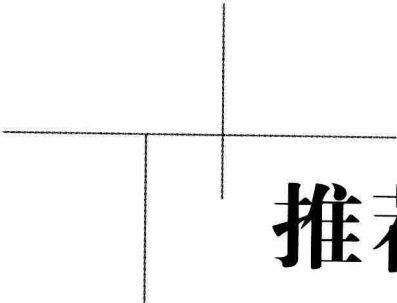
印 次:2016年12月第1次印刷

定 价:89.00元

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010)88254888,88258888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式:010-51260888-819 faq@phei.com.cn。



推荐序

这是一个最好的时代，我们赶上了互联网的大潮；这也是一个最坏的时代，我们习惯的旧秩序被互联网打破了；这是一个创新的时代，任何奇思妙想都有被广大人民接受成为现实的可能；这还是一个变革的时代，在“互联网+”的春风中，线下的思维转换为线上的思维。

这一切的推手，就是互联网。互联网催生了一种新型的生活方式，也催生了适应于互联网的一种新的人群。连接人群和线上生活的纽带就是一个一个的互联网及移动互联网的应用。作为应用的典型 IT 形态，互联网应用经历了独占型应用，SOA 应用，在云时代就是云原生（Cloud Native）的应用。构建一个良好的可伸缩的应用，不仅仅是一个优秀工程师的职责，同时也代表了对一种生活方式的认可。在这种应用里，我们看到了电商应用的身影：任何一个电商购物狂欢节，背后是工程师制作的良好弹性应用面对使用量波峰波谷的挑战。

除了电商应用，任何一个 To C 的 APP 都会面临同样的挑战，要承载捉摸不定的业务量及快速推进版本的演进。作为引来的昂贵的流量，需要良好的后台承接并及时处理，推动良好的客户体验及带来真实的业务发展，从而帮助 C 端的创业者快速验证自己的商业模式并快速迭代。

本书就是这样一本很好的最佳实践。本书作者将自己的成功经验总结归纳出来，对弹性架构的概念，软件设计的原则，以及如何构建一个优质的互联网应用，进行了深入讲解。作为一位互联网电商的资深从业者，书中很多概念与电商行业的最佳实践不谋而合，有些

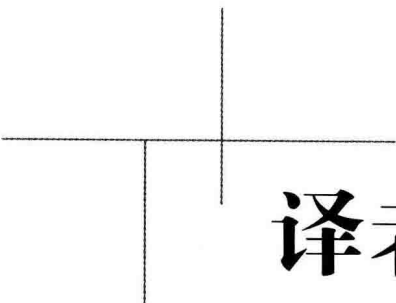
观点对电商网站的构建是非常好的建设性意见。

本书的译者李智慧作为互联网老兵，对互联网应用网站的架构设计有丰富的经验；同时作为大数据行业的资深从业者，对数据运营的观念理解得精准到位，结合互联网网站设计及数据运营的思路，推动了本书翻译的成稿，不仅对原著内容的理解到位，同时对原著细节的拔高有神来之笔。

本书适合互联网技术从业人员阅读，是电商行业、移动互联网行业中不可多得的好书。

杨海明

京东云平台事业部总经理



译者序

每个互联网创业者的心中都有一个独角兽或者纳斯达克的梦想,不过不论梦想有多么远大,都需要从第一行代码开始,一步一步地构建系统。那么,第一行代码和最后服务数亿用户的系统之间是什么关系呢?一个最初由两三个工程师开发的产品雏形,如何经过逐渐地重构、演化、迭代、伸缩,最终成为一个巨无霸的系统?

本书面向互联网创业公司工程师,讲述构建可伸缩 Web 系统的核心概念、关键原则和主要方法。本书第 1~5 章和第 9 章由李智慧翻译,第 6~8 章由何坤翻译。第 1 章和第 2 章讲述可伸缩系统的核心概念与软件基本设计原则。我强烈建议你认真阅读这两章,这部分内容包含了开发一个可伸缩的 Web 系统甚至开发一个良好软件的基本原理和设计原则,是其他一切技巧和方法的元规则。第 3~8 章讲述了构建可伸缩 Web 应用的主要方法和工具,这几章的内容相对比较独立,你可以挑选自己感兴趣的章节阅读,也可以在工作中遇到问题时快速浏览,寻找方法和灵感。第 9 章讲述可伸缩的系统运维及可伸缩的个人和团队,如果你正处在一个高速发展的创业团队中,如果你对从技术走向管理感兴趣,我相信你可以从本章的内容中收获多多。

诚如书中一再强调,可伸缩系统的设计是一种权衡的艺术,必须对每一种方案的优缺点都了如指掌,才能在面对实际问题时做出最合适的选择。高并发可伸缩系统的设计看似纷繁复杂庞大无比,实际上关键的核心技术也就那么几样,如果深入掌握了这些关键技术,就抓住了可伸缩系统设计的核心。这几样关键技术,可能需要在不同场景,从不同视角反复思考琢磨,才能真正掌握。我曾经在《大型网站技术架构:核心原理与案例分析》一书

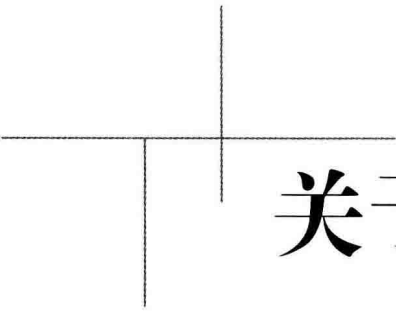
中表达过这个观点，如果读过《大型网站技术架构：核心原理与案例分析》，那么推荐你阅读本书，从创业公司工程师的视角再次审视分布式网站设计的方方面面，掌握可伸缩 Web 应用设计权衡的艺术。

虽然罗马不是一天建成的，但是一名优秀的工程师，在放下建设罗马的第一块砖的时候，心中就已经有了整个罗马建成后辉煌的场景，以及整个建造历程。他运筹帷幄、胸有成竹，每一次的快速发展他都已经提前做好准备，每一次的剧烈变化他都有应对的方案。因为对艰难困苦了然于心，所以倍加从容。

本书就是这样一本书，它描述了一个愿景，当你的公司成为独角兽公司的时候，你的系统会长什么样子；它展现了一个历程，你该如何一步一步地将你的系统伸缩成一个巨无霸系统；它提供了一组工具和方法，你如何利用这些工具方法改造你的系统实现你的梦想。

星空就在你的头顶，路就在你的脚下，梦想就在你的前方，带上这本书，我们出发吧！

李智慧



关于作译者

关于作者

Artur Ejsmont 是一位充满激情的软件工程师及敏捷领导者，目前就职于雅虎悉尼。Artur 从事 Web 应用方面的开发工作已经超过十年，同时也大量关注敏捷及创业公司管理，热爱精益创业模型。Artur 开发的网站每小时 PV 数超过一百万，为欧洲及澳洲两个大洲的用户提供服务。他还在大学开设课程，分享他的技术与知识。作为一名科技作家、博客写手、技术演讲者，Artur 对技术社区有许多积极的贡献。

关于技术编辑

Bill Wilder 是 Finomial 公司的 CTO，偶尔在技术会议上演讲，也是一位不连续更新的博客写手，波士顿 Azure 的领导者（从 2009 年开始）。他被微软授予 Azure MVP（最有价值专家）殊荣，著作有《云架构模式》（2012 O'Reilly Media 出版）。你可以通过 Twitter 账号@codingoutloud 关注他。

TJ Wilder 是马萨诸塞艾姆赫斯特学院的学生，主修计算机科学，辅修数学。他从 12 岁开始编程，偶尔会被其他事情打断，比如吃饭、睡觉、唱歌。当 DraftKings.com 还是创业公司的时候，他就在那里实习。他期待本书能给那些在创业公司工作的人们更多帮助。

Dr. Danny Coward 是 Oracle 的首席架构师及 Web 架构师。Coward 是 Java EE、Java SE/JavaFX Web Sockets 的 Java API 的独立专家领导，也是 Oracle WebSocket 技术文档的独立作者。Coward 在 Oracle 的 WebSocket 方面的领导工作使他成为 Java WebSocket 编程方面的领袖级专家。Coward 是一个有着十多年经验的 Java 开发者，擅长将复杂的技术简化以实现业务对象。Coward 在 Java 软件的所有方面（从 Java Me 到 JavaFx 技术）都有着专家级的经验。

关于译者

李智慧，曾供职阿里巴巴及英特尔亚太研发中心，从事分布式系统与大数据方面的开发。《大型网站技术架构：核心原理与案例分析》作者。目前正在互联网方面创业。

何坤，宅米首席架构师，前阿里巴巴平台架构师。曾参与阿里中文站架构建设，以及 WebX 框架、云计算存储平台 Doris 等核心系统研发。曾就职丰联金融证券期货部任首席架构师。对大型互联网站架构有深刻理解，对中间件、分布式系统及新技术研发始终保持热情。目前关注自然语言处理、机器理解等新课题。

致谢

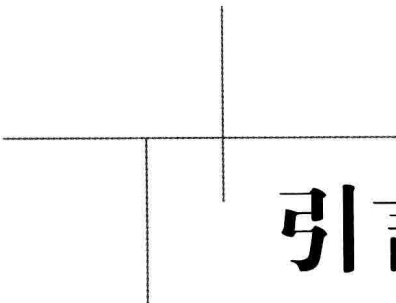
感谢 Bill Wilder、TJ Wilder、Danny Coward，以及 Brandi Shailer 在我写作本书期间给予的反馈和帮助。也感谢 Dion Beetson、Craig Penfold、Jackie Reses、Venkatesh Kanchan、Marcelo Maidana、Kristian Kauper、Christof Mueller、Ścibór Sobieski、Mariusz Jarocki、Andrzej Nowakowski，我的朋友们和家人们在过去 35 年间给予的支持和鼓励。没有他们就不会有我现在的成就。

本书献给所有热爱编程的你们，

Geek 们，

人类的未来真的就在你们手中，

就在此刻。



引言

近些年来，伴随着技术的进步，越来越多的 Web 应用系统需要存储、转化、处理越来越多的数据，而这必将要求工程师们掌握构建可伸缩的 Web 系统的能力。

当我了解到大多数工程师都缺乏这种构建可伸缩 Web 系统的能力时，我觉得有必要写一本与此有关的书。一方面，目前市面上缺乏相关的著作；另一方面，那些在小公司工作的工程师们也缺乏必要的环境去学习可伸缩架构的设计方法。因此，本书致力于讲解软件架构与基础设施如何协同工作，并最终实现系统的可伸缩性。

希望本书可以成为开启读者可伸缩架构设计之旅的一个里程碑。书中既给出了可伸缩架构设计的整体概览视图，同时也深入探讨了一些重要的技术点并给出了一些最佳实践建议。然而，书中并没有对每一个技术细节都深入阐述，而是尽可能地为读者呈现那些必要的概念、基本的规则，以及有意义的实例。

第 1 章：核心概念

作为本书第 1 章，本章主要介绍可伸缩性架构的一些核心概念及关于本书的一个概览性描述。本章通过描述可伸缩 Web 系统演化的不同阶段介绍相关的概念，同时，也会涉及一些基础设施与可伸缩 Web 系统架构方面的整体描述。

第 2 章：软件设计原则

本章主要探讨一些构建弹性可伸缩系统的设计原则。首先是那些最基础的概念，比如简化与解耦；其次是一些面向对象的设计原则，比如单一职责原则和依赖注入原则；最后

是那些与可伸缩性直接相关的设计概念，比如功能分割、数据分区，以及自我恢复。

第 3 章：构建前端层

本章内容主要集中在直接与客户端软件（比如 Web 浏览器及移动 APP）交互的那部分基础设施上。本章会深入解释前端管理状态的几种方式，以及构建可伸缩前端的几种重要组件，比如负载均衡、代理、内容分发网络（CDN）；之后会探讨自动化伸缩及几种部署场景。

第 4 章：Web 服务

本章主要探讨几种不同的 Web 服务架构的优缺点。解释 Web 服务设计原则并深入细节讲解基于 REST-ful API 的伸缩性技术。

第 5 章：数据存储层

本章会解释数据层伸缩性技术的若干核心要点。除了涉及 MySQL 这一类关系数据库相关的技术外，还会花大量篇幅探讨 Cassandra 这类 NoSQL 数据存储技术。本章会穿插讲解一些技术细节，比如数据分区（也常称作分片技术）、数据复制，以及最终一致性。比较几种不同的数据存储层拓扑结构及与此有关的技术挑战。

第 6 章：缓存

本章内容主要着眼于缓存，缓存是一种改善 Web 系统伸缩性和性能的关键性手段。主要讲解基于 HTTP 的几种不同的缓存技术，以及 HTTP 缓存的伸缩性技术。此外，还会阐述对象缓存技术及其常用的伸缩性技术。最后，花一些篇幅探讨缓存最佳实践，以帮助读者在使用缓存时做出更好的决策。

第 7 章：异步处理

本章会讲解有关消息与事件驱动架构方面的主题。首先，探讨异步处理的概念与优点及如何利用消息代理改善 Web 应用的可伸缩性；其次，重点论述异步系统的挑战与使用中可能出现的陷阱；最后，简要比较几种主流的消息平台以帮助读者在具体工作中做出最佳选择。

第 8 章：数据搜索

本章主要讨论数据搜索方面的有关问题。数据搜索与数据存储密切相关，随着数据集的快速增长，优化数据搜索与访问方面的技术变得越来越重要。首先，会讲解不同类型的索引是如何工作的；其次，会花一些篇幅描述数据建模如何有助于改善伸缩性，以及如何在诸如 Cassandra 之类的 NoSQL 数据存储系统中考虑数据建模。最后，做一些搜索引擎方面的介绍及讲述相关技术如何应用于 Web 应用系统。

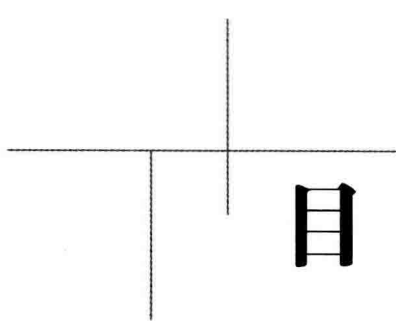
第 9 章：伸缩性的其他维度

这是本书的最后一章，阐述一些扩展读者自身工作效率的有关概念，以方便读者更好地实现自我管理 with 团队成长。首先，强调自动化作为一项关键技术 in 提高工程效率方面的重要作用，探讨有关自动化测试、部署、监控与报警方面的话题。其次，分享一些项目管理方面的个人经验与观察，这些技能将帮助读者在创业中更好地生存下来。最后，对成长中的敏捷团队可能遇到的挑战做一些反思。

目标读者

本书的主要目标读者是软件工程师、技术经理、DevOps，以及系统工程师。对于在校学生而言也许会有有一定难度，不过对于中等水平甚至是初级工程师而言，绝大多数内容都是比较容易理解的。

本书假设读者对于如何利用相关技术构建一个 Web 应用系统有一个基本的了解。不过阅读本书不需要拥有任何特定的编程技能，诸如 Java、PHP、JavaScript、C#或者 Ruby，因为伸缩性是 Web 应用开发中一个通用的挑战，与具体语言无关。另外，本书也假设读者了解 HTTP 协议是如何工作的，以及对 IP 网络、HTML、C/S（客户端/服务器）软件开发等有一个基本的概念。



目 录

1	核心概念	1
什么是伸缩性.....		2
从单一服务器到全球用户的 Web 架构演化.....		4
单一服务器.....		5
使用更强的服务器：垂直伸缩.....		6
服务分离.....		10
内容分发网络：静态内容的伸缩性.....		12
分散访问流量：水平伸缩.....		13
服务全球用户的伸缩性架构.....		16
数据中心基础设施架构概览.....		18
前端.....		19
Web 应用层.....		20
Web 服务层.....		20
附加组件.....		21
数据持久层.....		21
数据中心基础架构.....		22
应用架构概览.....		23
前端.....		24
Web 服务.....		25
支撑技术.....		29
小结.....		30

2

软件设计原则

31

简单.....	31
隐藏复杂与构建抽象.....	32
避免过度设计.....	33
尝试测试驱动开发.....	34
从软件设计的简化范例中学习.....	35
低耦合.....	36
促进低耦合.....	37
避免不必要的耦合.....	39
低耦合范式.....	40
不要重复自己 (DRY)	41
复制粘贴代码.....	42
基于约定编程.....	43
画架构图.....	46
用例图.....	49
类图.....	50
模块图.....	51
单一职责.....	52
改善单一职责.....	52
单一职责的例子.....	53
开闭原则.....	53
依赖注入.....	55
控制反转 (IOC)	57
为伸缩而设计.....	59
增加副本.....	60
功能分割.....	62
数据分片.....	63
自愈设计.....	65
小结.....	67

3	构建前端层	69
	状态管理.....	70
	管理 HTTP 会话.....	73
	管理文件.....	77
	管理其他类型的状态.....	80
	可伸缩的前端组件.....	83
	DNS.....	84
	负载均衡器.....	85
	Web 服务器.....	92
	缓存.....	93
	自动伸缩.....	94
	部署案例.....	96
	AWS 场景.....	97
	私有数据中心.....	98
	小结.....	101
4	Web 服务	102
	Web 服务设计.....	102
	Web 服务作为一种备用表示层.....	103
	API 优先方式.....	105
	务实的方式.....	107
	Web 服务类型.....	108
	以功能为中心的服务.....	109
	以资源为中心的服务.....	111
	伸缩 REST Web 服务.....	115
	保持服务无状态.....	115
	缓存服务响应.....	121
	功能分割.....	124
	小结.....	127

5 数据存储层 129

MySQL 伸缩性.....	130
复制.....	130
数据分区（分片）.....	142
NoSQL 伸缩性.....	157
最终一致性.....	160
快速恢复增加可用性.....	164
Cassandra 拓扑结构.....	166
小结.....	170

6 缓存 171

缓存命中率.....	171
基于 HTTP 的缓存.....	173
HTTP 缓存头.....	174
HTTP 缓存技术类型.....	179
伸缩 HTTP 缓存.....	185
缓存应用对象.....	188
对象缓存的一般类型.....	189
伸缩对象缓存.....	194
缓存的经验法则.....	198
缓存整个调用栈.....	198
用户间缓存重用.....	199
从哪儿开始使用缓存？.....	201
缓存失效的困难.....	201
小结.....	203

7 异步处理 204

核心概念.....	204
同步处理的例子.....	205
异步处理的例子.....	208
购物类比.....	211