

Practical Load Balancing
Ride the Performance Tiger

实用负载均衡技术 网站性能优化攻略

- 网站负载均衡架构全揭秘
- 完美应对云环境及大数据的挑战
- 网站性能优化必备指南

[英] Peter Membrey [澳] David Hows [荷] Eelco Plugge 著
武海峰 陈晓亮 译



人民邮电出版社
POSTS & TELECOM PRESS

013038560

TURING 图灵程序设计丛书

TP393.07
230

Practical Load Balancing
Ride the Performance Tiger

实用负载均衡技术 网站性能优化攻略

[英] Peter Membrey [澳] David Hows [荷] Eelco Plugge 著
武海峰 陈晓亮 译



北航

C1646550

TP393.07
230

人民邮电出版社

北京

022880810

图书在版编目 (C I P) 数据

实用负载均衡技术：网站性能优化攻略 / (英) 门伯里 (Membrey, P.) , (澳) 豪斯 (Hows, D.) , (荷) 普拉奇 (Plugge, E.) 著 ; 武海峰, 陈晓亮译. -- 北京 : 人民邮电出版社, 2013. 5

(图灵程序设计丛书)

书名原文: Practical Load Balancing: Ride the Performance Tiger

ISBN 978-7-115-31463-5

I. ①实… II. ①门… ②豪… ③普… ④武… ⑤陈… III. ①网站—载荷—均衡—研究 IV. ①TP393.07

中国版本图书馆CIP数据核字(2013)第072616号

内 容 提 要

本书介绍了处理负载均衡问题的相关概念和工具, 说明了如何避免性能退化和服务器上的服务突然崩溃的风险, 阐述了单个服务器以及可以执行 cookie 插入或者改善 SSL 吞吐量的负载均衡器, 最后还探讨了云计算中的负载均衡。

本书适合对系统架构、性能维护感兴趣的初级、中级读者以及有经验的系统架构师和运维师。

图灵程序设计丛书

实用负载均衡技术：网站性能优化攻略

-
- ◆ 著 [英] Peter Membrey [澳] David Hows
[荷] Eelco Plugge
 - 译 武海峰 陈晓亮
 - 责任编辑 丁晓昀
 - 执行编辑 梁 薇
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
大厂聚鑫印刷有限责任公司印刷
 - ◆ 开本: 800×1000 1/16
印张: 14.75
字数: 341千字 2013年5月第1版
印数: 1-3 500册 2013年5月河北第1次印刷
 - 著作权合同登记号 图字: 01-2012-3533号

ISBN 978-7-115-31463-5

定价: 55.00元

读者服务热线: (010)51095186转604 印装质量热线: (010)67129223

反盗版热线: (010)67171154

站在巨人的肩上
Standing on Shoulders of Giants



www.turingbook.com

版权声明

Original English language edition, entitled *Practical Load Balancing: Ride the Performance Tiger* by Peter Membrey, David Hows, Eelco Plugge, published by Apress, 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705 USA.

Copyright © 2012 by Peter Membrey, David Hows, Eelco Plugge. Simplified Chinese-language edition copyright © 2013 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由Apress L.P.授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

译者序

移动设备的日益普及，促使移动应用也随处可及。周末，你出去和朋友聚餐，在餐厅里拍一张狼吞虎咽的吃相，分享到 SNS 网络；或是在吃饭的空档，多年不见的同学想留下彼此的联系方式，大家在“一、二、三”的呐喊声中一起“摇一摇”！这时，你的移动设备已通过无线网络和远端服务器进行了多次交互，而服务器要及时地处理这些负载。加之现在移动互联产品发展态势迅猛，不少移动应用都已拥有大批用户，所以移动互联网应用的发展离不开服务器处理负载能力的提升。如何提高服务器处理负载能力，无疑关系着众多用户的实际体验。

互联网技术发展至今，服务器非但没有被取代，反而借助于新的互联网服务形式而蓬勃发展。大型服务器昂贵的价格和高昂的迁移成本，促使大量的 Web 服务都采取了横向扩展的方式来提高负载处理能力和可用性。

这些使用方式的改变，并未撼动服务器的位置，足见服务器负载处理是所有互联产品重要的基石。

本书面向对 Web 应用后台支撑系统感兴趣的 Web 开发和设计人员。全书围绕着横向扩展这个主题，力图准确到位地介绍基础概念及扩展服务器时所需要的具体工具和详细操作步骤，并在必要时补充大量的背景知识，以便读者迅速了解相关技术。

在翻译过程中，我们也看到了原书的诸多不足之处：垂直深度略显不足（书中并未介绍进行大规模服务器配置部署时的相关自动化工具，如 chef、puppet。）、介绍规划一章略显冗余（这些概念完全可以渗入其他章节，无需浪费篇幅）、原书看起来有点绕口（可能是因为合著的原因，很多地方对同个概念多次重复，甚至前后两段出现重复的几句话，颇显啰嗦），以及一些概念性错误（比如对虚拟内存的理解存在偏差，对 IPv6 的理解也出现了前后不一致的情况）。这些我们都竭尽所能，一一校正，还望与读者诸君交流为盼。

从整体上来看，这是一本比较好的负载均衡入门书籍，内容也较新（已出版的几本相关英文著作都较早）。鉴于国内尚未有同类译著上市，我们希望本书的出版能满足入门人员的需要。翻译过程中，我们针对其中涉及的概念和术语进行了多次讨论和推敲，并对译稿进行了严格的相互审读。囿于经验，难免有一些疏漏和不足之处，还需各位读者予以积极反馈。我们会在图灵社区维护一份勘误表（<http://www.ituring.com.cn/book/1050>），以便各位参考和在下一印次中更正。

武海峰，于杭州

前言

尽管 IT 行业对负载均衡、缓存以及性能的需求仍然强劲，但仍有人认为，这些技术实现起来要么极其昂贵，要么极其复杂，要么二者兼而有之。这或许并不难理解，毕竟不久前，只有少数大公司或专业公司才需要认真考虑这一问题。但在今天，即使一个个人博客也可能流量很大，导致服务器应付不过来。而公司网站则需要保证全天候运转，且不管什么情况每天如此。如果你的网站需要做到一直快速响应并且可靠运行，而你又不知道从何下手，或者手里没有几百万美金的预算（或者即使你有这个预算！），那么本书可以帮助你。

目标读者

本书面向那些建过网站，并想了解如何扩展网站来支持更多用户访问的读者。性能和可靠性的圣杯并不应局限在那些有充裕预算的大公司手上。只要掌握了本书介绍的技术，所有人都可以大幅提升应用的性能。

本书也面向希望推出下一个出色应用的读者，帮助他们确保所构建的应用能够应付不计其数的用户访问的压力。

本书结构

本书按传统的三部曲划分。首先介绍负载均衡和缓存技术背后的重要概念，讲解为什么要关注它们。在第一部分，我们讲述基础概念。我们会了解网站的运作机制：将各部分一一分解，逐一介绍，让读者了解它们之间如何交互，各部分在整体中的重要性。接下来用很大篇幅介绍缓存技术。我们会回顾它的历史，了解大量不同的缓存技术，以及如何使用其中的一些关键技术。之后阐述如何将 DNS 用作简单的负载均衡系统，如何利用内容分发网络搭建并运行网站（以及什么是内容分发网络和为什么要使用内容分发网络）。最后讨论项目管理和备份以及如何规划，从而获得优异的性能。

从第二部分开始，你将会了解更多实践内容。在第 7 章中，你会了解应对性能方面挑战的必要知识。第 8 章中，我们将会演示如何对网站服务器进行负载均衡。第 9 章沿着这个思路继续介绍如何对数据库进行负载均衡。第 10 章将会演示如何对网络自身进行负载均衡。第 11 章介绍如何对 SSL 连接进行负载均衡。

在第三部分中，我们将会超越常规的做法。第 12 章介绍如何搭建高可用性系统，第 13 章讨论云及其对与性能相关的应用的影响，第 14 章介绍 IPv6 和如何在 IPv6 平台上应用书中介绍的知识。第 15 章回顾书中的内容，并介绍一些相关的主题，比如安全、规划和操作系统中的性能提升。

下载代码

在 www.apress.com 网站本书主页 (www.apress.com/9781430236801) 上的 Source Code/Downloads 标签页里有源码下载链接。^①

^① 读者也可以在图灵社区本书页面下载源代码，网址是 ituring.com.cn/book/1050。

特别致谢 serverlove

在此，我们特别感谢 Daniel Keighron-Foster 和 serverlove 团队，他们慷慨地允许我们在写作本书时随意使用他们的云计算平台。

多年以来，我们一直是墨尔本服务器托管服务的忠实用户。他们提供的支持非常完善，服务器设施也很先进，并且真正关注客户的需求。要知道，你是难得碰到此类情况的：总经理能在半夜帮你把 2U 服务器抬到机架上，支持人员比你还要关心你所需处理的问题。不管什么时候打电话过去，我们总能得到满意的答复。

所以，当他们推出云解决方案时，我们毫不犹豫地就选用了。现在，我们在 serverlove 平台上托管着几个运营中的网站和应用，所以在寻找一个构建和测试本书中的解决方案的环境时，我们不加思索地打电话联系了他们。

你可能也猜到了，我们强烈推荐使用 serverlove。如果你想了解更多有关该平台的信息，请访问其官方网站 www.serverlove.com。我们还想说明的是，我们写这份致谢绝不是因为得到贿赂、受到胁迫或因其他非正常手段驱使。所有的溢美之词都源自服务器能够顺畅地运行并保持极高在线时间的实际情况。

致 谢

毋庸置疑，没有太太的关爱、支持和理解，这本书不可能完成。我已经记不清有多少次，我会突然看到一杯热咖啡置于书桌之上，或者在我盯着屏幕思索为什么服务器突然没反应时，她会安静地陪伴着我。全部的成功与所得，自然都有她的一份功劳。

另外，如果没有 Eelco 和 Dave 的加入，这本书也定然无法完成。他们投入到本书写作上的热情和努力令我自惭形秽。非常感谢他们，正是有了他们，本书才可能完成。

写作需要投入大量的精力和幕后的工作，所以在此，我要感谢 Brendan、Richard 和 Wouter 的专业审校，以及 Apress 出版团队的理解和包容。

最后，非常感谢 serverlove 的 Dan Keighron-Foster 和其他所有为完成本书作出过贡献的人。

——Peter Membrey

非常感谢咖啡与糖的组合，它们将我从疲惫中唤醒，赐予我无尽的活力。感谢 Pete 和 Dave 的辛勤和专注。感谢诸多手册页和文档的作者，为了研究这些东西，我睡得更晚了。

——Eelco Plugge

感谢 Kappallan Nallasami 先生无意间为本书提供一些宝贵意见——若无这些见解，就不会有这本书。

感谢 Joshua Ryan 和 Alexey Chernyak 传授系统性能优化方面的知识，并继续留在该领域工作——没有他们的工作，就不会有本书。

最后，感谢 Peter Membrey 邀请我们加入本书的写作，并带领我们完成了本书。

——David Hows

欢迎加入

图灵社区 ituring.com.cn

——最前沿的IT类电子书发售平台

电子出版的时代已经来临。在许多出版界同行还在犹豫彷徨的时候，图灵社区已经采取实际行动拥抱这个出版业巨变。作为国内第一家发售电子图书的IT类出版商，图灵社区目前为读者提供两种DRM-free的阅读体验：在线阅读和PDF。

相比纸质书，电子书具有许多明显的优势。它不仅发布快，更新容易，而且尽可能采用了彩色图片（即使有的书纸质版是黑白印刷的）。读者还可以方便地进行搜索、剪贴、复制和打印。

图灵社区进一步把传统出版流程与电子书出版业务紧密结合，目前已实现作译者网上交稿、编辑网上审稿、按章发布的电子出版模式。这种新的出版模式，我们称之为“敏捷出版”，它可以让读者以较快的速度了解到国外最新技术图书的内容，弥补以往翻译版技术书“出版即过时”的缺憾。同时，敏捷出版使得作、译、编、读的交流更为方便，可以提前消灭书稿中的错误，最大程度地保证图书出版的质量。

优惠提示：现在购买电子书，读者将获赠书款20%的社区银子，可用于兑换纸质样书。

——最方便的开放出版平台

图灵社区向读者开放在线写作功能，协助你实现自出版和开源出版梦想。利用“合集”功能，你就能联合二三好友共同创作一部技术参考书，以免费或收费的形式提供给读者。（收费形式须经过图灵社区立项评审。）这极大地降低了出版的门槛。只要有写作的意愿，图灵社区就能帮助你实现这个梦想。成熟的书稿，有机会入选出版计划，同时出版纸质书。

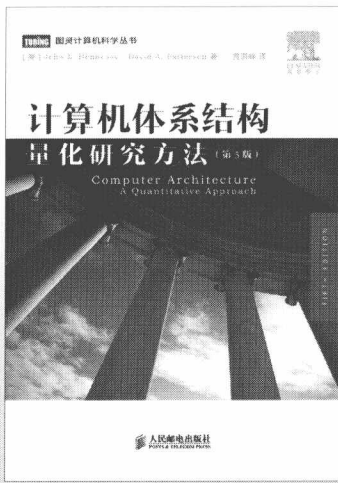
图灵社区引进出版的外文图书，都将在立项后马上在社区公布。如果你有意翻译哪本图书，欢迎你来社区申请。只要你通过试译的考验，即可签约成为图灵的译者。当然，要想成功地完成一本书的翻译工作，是需要有坚强的毅力的。

——最直接的读者交流平台

在图灵社区，你可以十分方便地写文章、提交勘误、发表评论，以各种方式与作译者、编辑人员和其他读者进行交流互动。提交勘误还能够获赠社区银子。

你可以积极参与社区经常开展的访谈、乐译、评选等多种活动，赢取积分和银子，积累个人声望。

试读结束，需要全本请在线购买：www.ertongbook.com



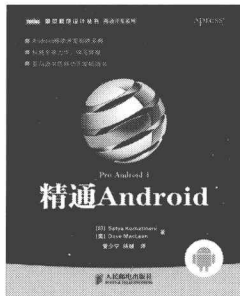
- ▶ 计算机体系结构的“圣经”
- ▶ 与时俱进的教科书
- ▶ 全新的第6章，讨论仓库级计算机

本书堪称计算机体系结构的“圣经”，是计算机体系结构方向的学生们的必读教材。全书系统地介绍了计算机系统的设计基础、指令集系统结构、流水线与指令级并行技术、层次化存储系统与存储设备、互连网络以及多处理器系统等重要内容，讲述了使用“量化研究方法”进行计算设计，以及多种可以实现并行的技术。在新版本中，增加了涉及移动中心和数据中心的示例，反映了在移动设备和云计算方面的变化；扩展和改进了多核和CPU体系结构方面的内容；增加了关于 warehouse-scale 计算机以及向量处理器和CPU的内容等等。

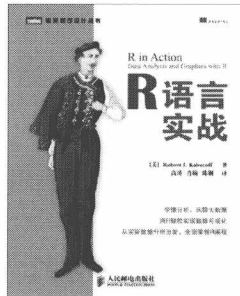
计算机体系结构：量化研究方法（第5版）
书号：978-7-115-29765-5
作者：John L. Hennessy David A. Patterson
定价：109.00元



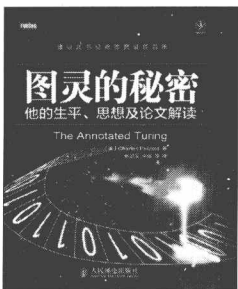
编译器设计（第2版）
书号：978-7-115-30194-9
作者：Keith D. Cooper
Linda Torczon
定价：99.00元



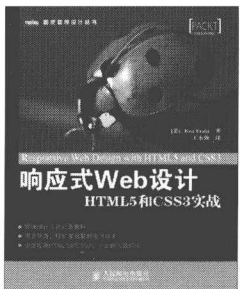
精通 Android
书号：978-7-115-29715-0
作者：Dave MacLean
Satya Komatineni
定价：119.00元



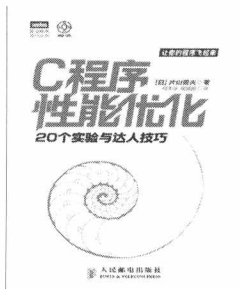
R语言实战
书号：978-7-115-29990-1
作者：Robert I. Kabacoff
定价：79.00元



图灵的秘密：他的生平、思想及论文解读
书号：978-7-115-28214-9
作者：Charles Petzold
定价：69.00元



响应式 Web 设计：HTML5 和 CSS3 实战
书号：978-7-115-29922-2
作者：Ben Frain
定价：49.00元



C 程序性能优化：20 个实验与达人技巧
书号：978-7-115-30000-3
作者：片山善夫
定价：29.00元

目 录

| | | | |
|---------------------|----|------------------|----|
| 第 1 章 引言 | 1 | 2.1.6 连接到数据库 | 12 |
| 1.1 性能问题 | 1 | 2.1.7 缓存技术速览 | 12 |
| 1.2 解决方案 | 2 | 2.1.8 回传到客户端 | 13 |
| 1.3 什么是负载均衡 | 3 | 2.2 进一步了解 | 13 |
| 1.3.1 负载均衡的前世 | 3 | 2.3 网络 | 14 |
| 1.3.2 负载均衡的今生 | 3 | 2.3.1 TCP | 14 |
| 1.3.3 纵向扩展 | 3 | 2.3.2 DNS | 16 |
| 1.3.4 横向扩展 | 4 | 2.3.3 速度、带宽和延迟 | 17 |
| 1.4 负载均衡的实现 | 4 | 2.3.4 网络连接小结 | 19 |
| 1.4.1 网络的构成 | 4 | 2.4 HTML 和 Web | 19 |
| 1.4.2 缓存：网站的曲速引擎 | 5 | 2.4.1 HTML | 20 |
| 1.4.3 使用 DNS 进行负载均衡 | 5 | 2.4.2 为什么基于文本很重要 | 20 |
| 1.4.4 内容分发网络 | 5 | 2.4.3 为什么链接很重要 | 21 |
| 1.4.5 6P 原则 | 6 | 2.4.4 HTML 小结 | 22 |
| 1.4.6 基础知识 | 6 | 2.4.5 浏览器 | 22 |
| 1.4.7 HTTP 负载均衡 | 6 | 2.5 Web 内容 | 23 |
| 1.4.8 对数据库进行负载均衡 | 7 | 2.5.1 静态内容 | 23 |
| 1.4.9 对网络连接进行负载均衡 | 7 | 2.5.2 动态内容 | 24 |
| 1.4.10 SSL 负载均衡 | 7 | 2.5.3 创建动态内容 | 24 |
| 1.4.11 建立高可用性集群 | 7 | 2.5.4 Web 内容小结 | 25 |
| 1.4.12 云平台上的负载均衡 | 7 | 2.6 数据库：最薄弱的环节 | 26 |
| 1.4.13 IPv6：实现和概念 | 8 | 2.7 总结 | 26 |
| 1.4.14 下一步做什么 | 8 | 第 3 章 内容缓存：保持低负载 | 28 |
| 1.5 总结 | 8 | 3.1 什么是缓存 | 29 |
| 第 2 章 网站工作原理 | 9 | 3.2 走马观花 | 29 |
| 2.1 开始我们的旅程 | 10 | 3.2.1 基于浏览器的缓存 | 29 |
| 2.1.1 来自非 IT 背景 | 10 | 3.2.2 Web 加速器 | 30 |
| 2.1.2 开始浏览的过程 | 10 | 3.2.3 Web 代理 | 31 |
| 2.1.3 通过 DNS 查找网站 | 11 | 3.2.4 透明 Web 代理 | 32 |
| 2.1.4 最终连接到服务器 | 11 | 3.2.5 边缘缓存 | 33 |
| 2.1.5 服务器自身 | 12 | 3.2.6 平台缓存 | 34 |

| | | | | | |
|-------|-------------------|----|-------|--------------------------|-----|
| 3.2.7 | 应用缓存 | 35 | 4.4.5 | 常见问题 | 63 |
| 3.2.8 | 数据库缓存 | 36 | 4.4.6 | 测试 DNS | 63 |
| 3.2.9 | 仅仅是个开始 | 36 | 4.5 | 基于 DNS 的负载均衡 | 64 |
| 3.3 | 缓存理论：缓存为什么这么难 | 36 | 4.5.1 | 基于 DNS 的负载均衡的优势 | 65 |
| 3.3.1 | HTTP 1.0 对缓存的支持 | 37 | 4.5.2 | 基于 DNS 的负载均衡的问题 | 65 |
| 3.3.2 | HTTP 1.1 加强的缓存支持 | 37 | 4.6 | 总结 | 66 |
| 3.3.3 | 解决方案 | 38 | 第 5 章 | 内容分发网络 | 67 |
| 3.3.4 | 缓存不像看起来那么简单 | 39 | 5.1 | 选择 CDN 服务提供商 | 68 |
| 3.4 | Web 代理 | 39 | 5.2 | 开始使用 Rackspace | 68 |
| 3.4.1 | Squid 代理服务器 | 39 | 5.3 | 向 CDN 账户添加内容 | 69 |
| 3.4.2 | 开始了 | 40 | 5.4 | Rackspace 云文件 API | 72 |
| 3.4.3 | 故障排除 | 41 | 5.4.1 | 将 API 集成到 PHP 中 | 72 |
| 3.4.4 | 透明代理 | 42 | 5.4.2 | 用 API 密钥进行认证 | 72 |
| 3.4.5 | 发生了什么 | 42 | 5.4.3 | 建立连接和断开连接 | 73 |
| 3.4.6 | 获得帮助 | 44 | 5.4.4 | 对容器进行操作 | 74 |
| 3.4.7 | Squid, 代理中的瑞士军刀 | 44 | 5.4.5 | 对文件进行操作 | 79 |
| 3.5 | 边缘缓存：Varnish | 45 | 5.4.6 | 其他有用的函数 | 84 |
| 3.5.1 | 默认保守缓存 | 46 | 5.5 | 总结 | 86 |
| 3.5.2 | 安装 Varnish | 46 | 第 6 章 | 性能和可靠性计划 | 87 |
| 3.5.3 | 配置并运行 | 47 | 6.1 | yoU MAke DIInner In TiME | 87 |
| 3.5.4 | 定制 Varnish | 49 | 6.1.1 | 理解 | 88 |
| 3.6 | 总结 | 49 | 6.1.2 | 决策 | 90 |
| 第 4 章 | 基于 DNS 的负载均衡 | 51 | 6.1.3 | 设计与实现 | 91 |
| 4.1 | DNS 内幕 | 51 | 6.1.4 | 安装 | 92 |
| 4.1.1 | IP 地址 | 51 | 6.1.5 | 测试、维护、评估 | 93 |
| 4.1.2 | 问题 | 52 | 6.1.6 | 计划的重要性 | 95 |
| 4.1.3 | 解决方案 | 52 | 6.2 | 备份 | 96 |
| 4.1.4 | 回退一步 | 53 | 6.2.1 | 为什么备份如此重要 | 96 |
| 4.2 | DNS 详解 | 53 | 6.2.2 | 前方可能有麻烦 | 97 |
| 4.2.1 | 亲自查询 | 54 | 6.2.3 | 必须实现自动化 | 98 |
| 4.2.2 | DNS 查询进阶 | 55 | 6.2.4 | 战术备份 | 98 |
| 4.3 | DNS 缓存 | 56 | 6.2.5 | 战略备份 | 98 |
| 4.3.1 | 查询 DNS 缓存 | 56 | 6.2.6 | 增量备份与全备份 | 99 |
| 4.3.2 | Linux 系统上的 DNS 缓存 | 58 | 6.2.7 | 一定，一定要测试恢复！ | 99 |
| 4.3.3 | 实质内容 | 58 | 6.3 | 总结 | 100 |
| 4.4 | BIND9 | 58 | 第 7 章 | 负载均衡基础 | 101 |
| 4.4.1 | DNS DB 的头 | 60 | 7.1 | 什么是负载均衡 | 101 |
| 4.4.2 | DNS 数据库记录 | 61 | 7.2 | 有哪些可用的计算资源 | 102 |
| 4.4.3 | 加载数据库 | 62 | 7.2.1 | 处理器 | 102 |
| 4.4.4 | 检查配置文件 | 63 | | | |

| | | | | | |
|-------|--|-----|--------|---------------------|-----|
| 7.2.2 | 内存 | 103 | 9.1.8 | 测试上述安装和配置 | 133 |
| 7.2.3 | 使用 top 命令查看 CPU 和 RAM 的性能 | 103 | 9.2 | 实施负载均衡 | 135 |
| 7.2.4 | 网络 | 104 | 9.2.1 | 建立负载均衡 | 135 |
| 7.2.5 | 存储 (磁盘) | 105 | 9.2.2 | 设置负载均衡服务器 | 136 |
| 7.3 | 负载均衡实战 | 105 | 9.2.3 | 设置工作服务器 | 137 |
| 7.4 | 指导原则 | 106 | 9.2.4 | 测试负载均衡服务器 | 138 |
| 7.4.1 | 深入理解系统 | 106 | 9.3 | 总结 | 139 |
| 7.4.2 | 规划 | 106 | 第 10 章 | 对网络进行负载均衡 | 140 |
| 7.4.3 | 监测和测试 | 107 | 10.1 | 分担负载 | 140 |
| 7.5 | 总结 | 107 | 10.2 | TCP/IP | 141 |
| 第 8 章 | 对网站进行负载均衡 | 108 | 10.2.1 | TCP | 141 |
| 8.1 | 测量 Web 服务器的性能 | 108 | 10.2.2 | IP | 142 |
| 8.2 | 加速 Apache HTTP | 110 | 10.3 | 路由 | 142 |
| 8.2.1 | 禁用空载模块 | 111 | 10.4 | 负载均衡服务器 | 144 |
| 8.2.2 | 禁用 DNS 查询 | 111 | 10.5 | IPVS | 146 |
| 8.2.3 | 采用压缩 | 112 | 10.5.1 | IPVS 的调度方式 | 146 |
| 8.2.4 | FollowSymLinks 和 SymlinksIfOwnerMatch 选项 | 113 | 10.5.2 | 在 Ubuntu 上安装 IPVS | 148 |
| 8.3 | 加速 nginx | 114 | 10.5.3 | 在 CentOS 上安装 IPVS | 149 |
| 8.3.1 | worker_processes 和 worker_cpu_affinity | 114 | 10.6 | IPVSADM | 150 |
| 8.3.2 | Gzip 压缩 | 115 | 10.7 | 扩展 IPVS | 151 |
| 8.4 | 对 Web 服务器进行负载均衡 | 116 | 10.8 | IPVS 进阶 | 153 |
| 8.4.1 | 配置 | 117 | 10.8.1 | 修改调度算法 | 153 |
| 8.4.2 | 准备 IPVS 服务器 | 118 | 10.8.2 | 分配权值 | 154 |
| 8.4.3 | 准备工作服务器 | 120 | 10.8.3 | 协议与多台虚拟服务器 | 155 |
| 8.4.4 | 测试负载均衡器 | 121 | 10.8.4 | 增加 IP 地址 | 156 |
| 8.5 | 划分动态和静态内容 | 122 | 10.9 | 保存设置 | 158 |
| 8.6 | 总结 | 123 | 10.10 | 总结 | 159 |
| 第 9 章 | 对数据库进行负载均衡 | 124 | 第 11 章 | 对 SSL 进行负载均衡 | 160 |
| 9.1 | 搭建 MySQL Cluster | 124 | 11.1 | 什么是 SSL 和 TLS | 160 |
| 9.1.1 | 安装管理程序 | 126 | 11.2 | 公钥密码学 | 161 |
| 9.1.2 | 配置管理程序 | 126 | 11.3 | 信任和数字证书认证机构 | 162 |
| 9.1.3 | 准备集群数据节点 | 129 | 11.4 | TLS 加密 | 162 |
| 9.1.4 | 安装 MySQL Server 和 NDB 守护进程 | 129 | 11.5 | TLS 负载均衡 | 162 |
| 9.1.5 | 配置 NDB 守护进程 | 130 | 11.6 | 配置 Web 服务器上的 SSL | 163 |
| 9.1.6 | 启动集群节点上的服务 | 131 | 11.6.1 | 配置 Apache 服务器上的 SSL | 165 |
| 9.1.7 | 更新 MySQL 的 root 用户 | 132 | 11.6.2 | 配置 nginx 服务器上的 SSL | 166 |
| | | | 11.7 | SSL 加速 | 166 |
| | | | 11.7.1 | 在 Apache 上启用 SSL 加速 | 166 |
| | | | 11.7.2 | 在 nginx 上启用 SSL 加速 | 166 |
| | | | 11.8 | SSL 前端 | 167 |

| | | | | | |
|-------------------------|---------------------------------|-----|---------------------------|------------|-----|
| 11.9 | 测试 SSL | 168 | 13.5 | 云的弹性 | 200 |
| 11.10 | 进一步配置 | 170 | 13.6 | 用云服务器工作 | 201 |
| 11.10.1 | 在 SSL 前端中启用 SSL 加速 | 170 | 13.7 | 总结 | 203 |
| 11.10.2 | 启用缓存 | 171 | 第 14 章 IPv6: 影响和概念 | 204 | |
| 11.10.3 | 指定要支持的协议 | 172 | 14.1 | IPv6 | 204 |
| 11.10.4 | 指定加密方法 | 172 | 14.2 | 十六进制表示 | 204 |
| 11.11 | LVS 和 SSL 终结前端 | 173 | 14.3 | 缩略表示 | 205 |
| 11.12 | 将负载均衡服务器/SSL 终端功能 集成到同一台服务器上 | 175 | 14.4 | IPv4 地址的耗尽 | 205 |
| 11.13 | 总结 | 175 | 14.5 | 部署 IPv6 | 205 |
| 第 12 章 使用集群提高可用性 | 176 | | 14.6 | IPv6 的优势 | 206 |
| 12.1 | 高可用性 | 177 | 14.7 | 实现 | 206 |
| 12.2 | 单一故障点 | 178 | 14.8 | 互联网连接 | 208 |
| 12.3 | 集群化 | 179 | 14.9 | DNS | 208 |
| 12.4 | IPVS 故障恢复 | 180 | 14.10 | 操作系统 | 208 |
| 12.4.1 | 在 Ubuntu 上安装集群软 件包 | 181 | 14.11 | 网络 | 208 |
| 12.4.2 | 在 CentOS 上安装集群软 件包 | 182 | 14.11.1 | 单一网关的网络 | 209 |
| 12.4.3 | 配置集群 | 182 | 14.11.2 | 双重网络 | 209 |
| 12.4.4 | 常见配置问题 | 185 | 14.12 | 软件支持 | 209 |
| 12.4.5 | 检查系统 | 186 | 14.12.1 | Apache | 209 |
| 12.5 | 测试 | 187 | 14.12.2 | nginx | 210 |
| 12.6 | Web 服务器细节配置 | 189 | 14.12.3 | Varnish | 210 |
| 12.6.1 | Ubuntu | 189 | 14.12.4 | Memcached | 210 |
| 12.6.2 | CentOS | 189 | 14.12.5 | IPVS | 211 |
| 12.7 | 高级配置选项 | 189 | 14.12.6 | ldirectord | 211 |
| 12.7.1 | ha.cf | 189 | 14.12.7 | heartbeat | 211 |
| 12.7.2 | ldirectord.cf | 190 | 14.13 | 总结 | 212 |
| 12.7.3 | Web 服务器 | 190 | 第 15 章 何去何从 | 213 | |
| 12.8 | 总结 | 190 | 15.1 | 回顾 | 213 |
| 第 13 章 云端负载均衡 | 191 | | 15.2 | 监控 | 214 |
| 13.1 | 云计算 | 191 | 15.3 | 安全 | 215 |
| 13.2 | 虚拟化 | 192 | 15.3.1 | 访问控制 | 215 |
| 13.3 | 虚拟化资源 | 195 | 15.3.2 | 视图 | 216 |
| 13.4 | 管理虚拟资源 | 196 | 15.3.3 | 常见的攻击防护 | 216 |
| 13.4.1 | 平衡 | 197 | 15.4 | 操作系统性能 | 217 |
| 13.4.2 | 超量供给 | 198 | 15.4.1 | 自己编译 | 217 |
| 13.4.3 | 计划 | 199 | 15.4.2 | 裁剪 | 218 |
| | | | 15.4.3 | 高性能操作系统 | 218 |
| | | | 15.5 | 计划 | 218 |
| | | | 15.6 | 总结 | 219 |
| | | | 索引 | 220 | |



互联网，尤其是万维网，为分布在世界各地的企业和个人提供了一个公平的竞争环境。你有建站或提供互联网服务的好点子吗？即使是比较前卫的点子，也能在启动资金不多的情况下以相对简易、低廉的方式实现。每个月花一点钱，你就能享受共享托管服务了，而独立主机（虚拟主机或者非虚拟主机）的价格甚至比几年前的托管主机还要便宜。

这样当然好了，可以放手去干而不必担心囊中羞涩也是一种进步。但如果你想让应用吸引更多的用户，该怎么办？如果它一炮走红，你发现手中的是下一个Facebook或者亚马逊，那时你又该怎么办？

1.1 性能问题

想象一下，你提出了下一个了不起的创意。它很棒、令人惊叹，而且今后几年里，人们会一直谈论它。你很确信自己会成为《时代》杂志的年度人物。毫无疑问，你很快就会和比尔·盖茨和马克·扎克伯格平起平坐。你的前途一片光明！

你花了几个月时间准备发布会。你进行了一遍又一遍的测试，邀请朋友和家人访问你的网站，还跑了一些beta测试。好评如潮！网站看起来很绚，用起来也很方便，每个人都很喜欢。你觉得再无改进空间了，决定立即发布。你伸手按下了那个神奇的按钮：网站上线了！然后坐下，等待属于你的赞美！

一开始，一切都还好。你得到了一些反馈，人们很满意，他们不停地拉朋友访问这个网站，因此注册用户越来越多。此时，你兴奋不已！

但好像什么地方出了问题。有人把网站链接发到Slashdot上了。瞬间，所有人都想看一下你的这个网站。服务器刚才还是每秒提供十几个页面，但现在正承受着每秒1万次页面访问的严峻考验！这完全出乎你的意料！不仅服务器并未按如此大的负载设计，而且你还注意到，数据库遭受冲击后，之前还能很快响应的请求也在变慢。一些人开始看到错误页面，一些人被提示访问超时，还有一些甚至不能看到这两个提示页面！

30分钟后，一切都完了。Slashdot报道说你的网站无法承受负载，每个人都在说这网站真烂。“一点都不稳定，”他们说，“有一半时间甚至都加载不了！”他们抱怨道：“完全是浪费时间！”还不停地发着牢骚。