Broadview
www.broadview.com.cn

构建高性能Web站点

Building High Performance Web

改善性能和扩展规模的具体做法

The Way to Improve Performance and Scale Out

郭欣 著

Smart Developer

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

构建高性能Web站点

Building High Performance Web

郭欣 著

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

本书围绕如何构建高性能 Web 站点，从多个方面、多个角度进行了全面的阐述，涵盖了 Web 站点性能优化的几乎所有内容，包括数据的网络传输、服务器并发处理能力、动态网页缓存、动态网页静态化、应用层数据缓存、分布式缓存、Web 服务器缓存、反向代理缓存、脚本解释速度、页面组件分离、浏览器本地缓存、浏览器并发请求、文件的分发、数据库 I/O 优化、数据库访问、数据库分布式设计、负载均衡、分布式文件系统、性能监控等。在这些内容中充分抓住本质并结合实践，通过通俗易懂的文字和生动有趣的配图，让读者充分并深入理解高性能架构的真相。同时，本书充分应用跨学科知识和科学分析方法，通过宽泛的视野和独特的角度，将本书的内容展现得更加透彻和富有趣味。

图书在版编目（CIP）数据

构建高性能 Web 站点 / 郭欣著. —北京：电子工业出版社，2009.8
ISBN 978-7-121-09335-7

Ⅰ. 构… Ⅱ. 郭… Ⅲ. 主页制作－程序设计 Ⅳ. TP393.092

中国版本图书馆 CIP 数据核字（2009）第 128794 号

策划编辑：李 冰
责任编辑：江 立
印 刷：北京智力达印刷有限公司
装 订：北京中新伟业印刷有限公司
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：787×980 1/16 印张：26.25 字数：608 千字
印 次：2009 年 8 月第 1 次印刷
印 数：4000 册 定价：59.00 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。

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

服务热线：（010）88258888。

谨以此书

献给我亲爱的父母和 Cris

推荐序

你很幸运能拿到这本书，更重要的是，你的网站用户也会很幸运。郭欣在这本书里深入而系统地分享了构建高性能网站技术的方方面面。从后台到前台，从网络传输到数据存储，涉及诸多技术原理和实现细节。通俗的语言，亲切的叙述，仿佛作者在耳边轻轻细语，然而又蕴含着一种技术思想和力量，并且融合了人文思想。

我曾经代表公司面试过许多开发人员，在问及与高性能相关的问题时，大家都能回答出需要负载均衡，需要缓存技术，然而当我进一步询问负载均衡如何实现或如何有效控制缓存命中率时，面试者却无从答起。知其然而不知其所以然是很多浮躁开发者的通病，也因此限制了其技术能力的提升和发展。

这本书将为你提供构建高性能网站的完整解决方案，它会成为每个致力于开发承载百万级用户规模网站开发者的工具箱。郭欣有着架构和开发多个大规模网站的经验，他精通前/后台技术和架构。在知道他将花时间著作一本高性能网站架构的书时，我不禁为国内许多开发者感到高兴。我见过部分知名网站架构师曾经分享过他们网站技术发展的历程，但每每都是停留在抽象层面，而像本书这样全面彻底地进行技术剖析却是头一回。尤其是构建高性能网站的各种技术方案，绝大部分是通过实践总结出来的经验，没有这样的经历，你甚至很难想象为什么会是这样。

不要犹豫了！当你拿起这本书，按照书中所分享的技术方案去实践时，你会发现，原来构建高性能网站就这么简单。中国互联网正在不断地成长，用户规模也在不断地扩大，我相信，越来越多的网站会根据性能这项最基本的用户体验决定其自身的生存能力。本书所提供的技术解决方案，正是在这个发展趋势中的一个基础，拥有它并加以实践，你和你的用户都会更加享受这一切！

为了页面一秒响应的境界，开始阅读吧！

——王速瑜

腾讯 R&D 研发总监（Tencent Director of R&D）

资深技术专家（Senior Technology Expert）

深圳，2009 年 7 月

专家评价

《构建高性能 Web 站点》是作者在 Web 系统领域多年工作、实践和探索的结晶。本书涉及 Web 系统优化的各个方面，从浏览器、Cache 到 Web、数据库和分布式文件系统等；穿插了大量的实际测试数据和很多流行开源软件的使用方法与案例；内容丰富，文字生动，对比形象。对于网络系统架构师、运维和开发人员，这是很好的参考书目；对于想了解 Web 性能并希望动手实践的人员，这是由浅入深的学习书籍。

——章文嵩博士，LVS 作者，Linux 内核作者之一

本书深入分析了常见的高性能 Web 技术的方法和原理，对搭建高性能 Web 站点具备很强的可操作性。

——张松国，腾讯网技术总监

这是一个令人兴奋的领域，这一系列准则和方法在 TopN 的互联网公司中都有大规模的实践和应用，作者在书中进行了详细而量化的论述。如果你正在为日益庞大的应用而手足无措，那么你唯一要做的就是拥有这本书，并且实践它。

——朱鑫，MemcacheDB 作者，新浪网研发中心平台部高级工程师

互联网寄托着我们的梦想，它改变了人们的生活，从社交网站到网络游戏，从搜索引擎到电子商务，成功的秘诀在于如何构建高性能 Web 站点。郭欣在这本书中几乎涵盖了 Web 性能优化的所有内容，并从多个角度进行了全面的阐述，你可以通过其通俗易懂的文字深入理解高性能站点架构的真相，并开拓视野，从而对性能瓶颈对症下药。本书可谓是高性能站点的必读精作。

——沈翔，Google Developer Advocate，加州总部

INTRODUCTION
前　言

从我写出第一个 HTML 网页到现在，已经过去 10 年多的时间了，回顾过去的 Web 开发经历，我曾经尝试过各种不同的技术，与此同时，我和我的团队也犯了很多的错误，但我们为此感到自豪。是的，成长是需要不断付出代价的，每次的挫折都会让我更加深刻地看到隐藏在深处的本质，为什么不把这些内容分享出来呢？于是便有了《构建高性能 Web 站点》这本书。

10 年来，我们见证了互联网有史以来最快速的发展，商业应用层出不穷，业务逻辑不断复杂，对用户体验的要求也不断提升，随之而来的是应用技术和开发语言的日新月异，开发者永不停息地学习新技术。同样，在 Web 站点性能方面，我们一直在跟时间赛跑，社交网站和微博客成为大众的主流应用，带来了更加快速、实时的信息传递，更多的站点意识到开放的重要性，数据访问和计算无处不在，每秒数以万次的数据传递和读写正在我们身边进行。

但是，构建 Web 站点的基础技术几乎多年来从未改变，比如诞生于 20 世纪 80 年代的 TCP，如今依旧是网络数据传输的主宰者，而 HTTP 则更与我们息息相关，可是你真的认真学习过它们吗？人们始终在做的事情就是在这些基础技术之上一层一层地封装概念，不断地诞生新的技术。加上商业化产品的市场竞争和炒作，.NET 和 Java 阵营中的概念让我眼花缭乱却又无可奈何。它们已经成为营销用语，有时候过度会让事情变得更加复杂，让开发者迷失方向。

不论你是一名从事 Web 开发的工程师，还是一名关心 Web 性能的架构师，都应该更多地关注各种技术和架构的本质。

从哲学意义上讲，对本质的研究属于形而上学的范畴，但是在自然科学中，我们从来不缺乏对本质的探索，因为只有认识事物的本质才能做出正确的决策，并且真正地驾驭它们，这是毫无争议的。

也许你曾经被商家的促销活动所打动。是的，我们往往只看到事物的表面现象，而经济学家却看到了事物的本质，这正是他们的高明之处。技术和架构同样如此，你要明白任何收获都是有代价的，天下没有免费的午餐，很多时候，你完全可以用成本经济学的知识来思考技术的合理性，你甚至可以像经济学家一样思考技术问题。

当然，仅仅理解本质是远远不够的，因为在庞大的架构体系中，涉及太多的部件，而影响整体性能的因素究竟有哪些呢？你也许会感到扑朔迷离，但你必须知道瓶颈所在，并且能够意识到何时需要优化性能或者扩展规模。与此同时，系统化的分析方法至关重要，中医理论对人体的系统思辨能力体现了先哲们的智慧，在站点性能不尽如人意的时候，我们能否“对症下药”？这与你对整个系统能否全面把握有着密切的关系。

另一方面，绝对与相对、变化与平衡，是永恒的大道，在很多时候你实际上需要考虑的是如何做出权衡，同时，我们也要铭记变化的道理，系统瓶颈不是一成不变的，久经考验的架构师深知这一点。

道可道，非常道。要将所有的架构之道讲出来实属不易，架构就像艺术品一样，往往无法完全复制，但是独立的技术以及分析的思路是可以学习的，作为优秀的开发者或者架构师，心中的架构才是最有价值的。

如果你希望寻找心中的架构，那么，从本书的绪论开始吧！

读者群

如果你希望学习如何创建一个 Web 站点，那么这本书可能并不适合你，但是当你对站点的性能开始担忧时，欢迎你的归来。

这本书适合以下读者：

- 编写 Web 应用程序、关心站点性能，并且希望自己做得更加出色的开发人员
- 关心性能和可用性的 Web 架构师
- 希望构建高性能 Web 站点的技术负责人
- 实施 Web 站点性能优化或者规模扩展的运维人员
- 与 Web 性能有关的测试人员

的确，整个技术团队的所有成员都适合阅读这本书。另外，高校学生以及个人网站站长也可以阅读，笔者希望本书可以帮助他们开拓视野。

如何阅读

本书涉及大量的软件和工具，由于篇幅有限，书中并没有对它们的具体使用方法展开详细的介绍，你可以通过 Google 查找相关的在线手册来进行自学，同时，在本书的参考文献列表和配套 Blog 中，也会提供一些相关链接。

本书不想在阐释体系上束缚读者的天分，所有内容的安排更像是引导你身处 Web 站点的

各个角落，与影响性能的各种因素自由碰撞，并且一步步地思考和分析问题。所以，你也许不会直接找到一个高性能 Web 站点的完整解决方案，但是，当你认真地依次阅读完本书的所有章节后，你也许会找到你更想要的东西，并从中获益，我衷心希望如此。

如果没有什么特殊的原因，还是建议你能够按照章节顺序，心平气和地通读一遍，因为在内容组织上，本书有太多的连贯性设计，我担心随机阅读会让你的收益大打折扣。

创作过程

说到这次写书的过程，我同样感到很有意义，各种全新的尝试让创作过程充满乐趣，它们同样值得分享。

不同于传统使用 Word 编写内容，我使用了快捷的 Google 在线文档，并使用在线表格保存测试数据，它们支持出色的版本管理，并且提供快速的分享和协作功能。当然，最激动人心的莫过于我可以在任何地点通过浏览器继续我的写作过程，甚至当灵感突如其来时打开 Google G1 手机便可以写上两句。

对于几十万字的篇幅，一气呵成绝对是不可能的，多次迭代必然贯穿整个写作过程，从灵感到提纲，再从框架到最终文字，虽然没有完善的过程管理，但是我时刻能感觉到敏捷的火花。

为了尽早地获得读者的反馈，我考虑尽早“部署”，于是选择了讨论组和邮件列表的方式，在 Google Group 上创建了读者讨论组，上传了一些试读章节，收集到了大量的修改意见和想法，这些都是我所需要的，同时也给我带来了鼓励和支持。

整个过程还有很多的花絮，这里就不一一介绍。创作的过程是艰辛的，需要作者的坚持和毅力，虽然创作本身没有捷径，但是我们可以让创作过程更加充满乐趣，让作者和读者更加近距离地接触。

当我将这些过程介绍给一些朋友时，他们感到很有意思，于是我们创立了 SmartDeveloper 系列，希望能够将这种敏捷写作过程进一步整理和完善，当然，《构建高性能 Web 站点》将作为该系列的开山之作。

SmartDeveloper 的具体内容敬请关注以下地址：

http://smartdeveloper.cn

值得一提的是，为本书撰写推荐序的王速瑜先生在敏捷开发领域有着丰富的经验，并在腾讯公司内部积极推广敏捷开发平台和方法。长久以来，我认为我们都是敏捷原住民，骨子里充满了敏捷的思想和战斗力。不可否认，敏捷给我带来了无法估量的收获。幸运的是，

他也计划写一本关于敏捷开发的书，总结他的实战经验，并且加入 SmartDeveloper 系列，我也非常期待这本书的问世。

延展阅读

由于篇幅和时间的限制，书中的部分内容点到为止，但是我也希望能够在未来继续和读者进行沟通，所以你也可以访问以下站点，来进一步关注其他的延展阅读资料。

http://highperfweb.com

读者讨论组

作为读者讨论和邮件订阅的最佳途径，我仍然选择了以下讨论组，在本书出版后的任何时间，你都可以来这里提出自己的意见和想法，我会对所有的主题给予回复。

http://groups.google.com/group/highperformanceweb

致谢

感谢我的父母，他们在我读初中的时候送给我第一台电脑（Cyrix1.66G 的兼容机），让我走进了计算机的世界，并且给予我非常多的支持和鼓励，让我毫无顾虑地追逐梦想。

感谢我的 Cris，她为我创作了本书的封面，并且在整个写作过程中毫无抱怨地陪伴我，给予我无尽的支持和灵感。

感谢电子工业出版社的策划编辑李冰和文字编辑江立，她们严谨认真的工作态度以及作为出版商的开放态度让我深感敬佩。

感谢为本书撰写推荐序的王速瑜先生，以及撰写评价的章文嵩博士、张松国先生、沈翔先生、朱鑫先生，他们在繁忙的工作中抽出时间，阅读了本书的样稿并写下了推荐和评价。特别一提的是，章文嵩博士对书中一些内容的理解和建议让我受益匪浅。

感谢对本书提出宝贵修改意见的朋友，他们是蒋琦、丁吉亮、汤文亮、刘健、朱李、周伟强。

感谢在读者讨论组中所有积极阅读试读内容并提出意见的成员。

CONTENTS

目 录

第 1 章
绪论

一般而言，人们评估一个 Web 站点的性能如何，通常先置身于用户的角度，访问该站点的一系列页面，体验等待时间。

当用户输入页面地址后，浏览器获得了用户希望访问该地址的意图，便向站点服务器发起一系列的请求，请注意，这些请求不光包括对页面的请求，还包括对页面中许许多多组件的请求，比如图片、层叠样式表（CSS）、脚本（JavaScript）、内嵌页面（iframe）等。接下来的一段时间，浏览器等待服务器的响应以及返回的数据。待浏览器获得所有的返回数据后，经过本地的计算和渲染，最终一幅完整的页面才呈现于用户的眼前。

1.1 等待的真相

整个过程听起来好像并不复杂，也许你从来都没有考虑过在这段等待的时间里世界都发生了什么变化，也许你早已习惯了利用这段时间东张西望或者品尝零食，或者你根本没有来得及意识到这点，新的网页就已经闪亮登场，恭喜你，你很幸运！但是在这个世界上，幸运儿永远只占少数，大多数人的大脑处理速度已经让他们明显感觉到这段等待时间漫长无比，久经考验的他们可以随时身手敏捷地打开多个浏览器窗口与时间赛跑，并为此筋疲力尽。

另一方面，对于站点经营者来说，让用户等待的时间过长，也许会造成毁灭性的后果。我见过很多人为了享用某家特色小吃而在餐馆门口乐此不疲地排着长队，但没有听说有多少用户执著地等待着一个速度缓慢的站点而不去尝试别的站点。

在这段等待的时间里，到底发生了什么？事实上这并不简单，大概经历了以下几部分时间：

- 数据在网络上传输的时间
- 站点服务器处理请求并生成回应数据的时间
- 浏览器本地计算和渲染的时间

数据在网络上传输的时间总的来说包括两部分，即浏览器端主机发出的请求数据经过网络到达服务器的时间，以及服务器的回应数据经过网络回到浏览器端主机的时间。这两部分时间都可以视为某一大小的数据从某主机开始发送一直到另一端主机全部接收所消耗的

总时间，我们称它为响应时间，它的决定因素主要包括发送的数据量和网络带宽。数据量容易计算，但是究竟什么是带宽呢？我们将在后续章节中详细介绍带宽的本质。

站点服务器处理请求并生成回应数据的时间主要消耗在服务器端，包括非常多的环节，我们一般用另一个指标来衡量这部分时间，即每秒处理请求数，也称吞吐率，注意这里的吞吐率不是指单位时间处理的数据量，而是请求数。影响服务器吞吐率的因素非常多，比如服务器的并发策略、I/O 模型、I/O 性能、CPU 核数等，当然也包括应用程序本身的逻辑复杂度等。这些将在后续章节中详细介绍。

浏览器本地计算和渲染的时间自然消耗在浏览器端，它依赖的因素包括浏览器采用的并发策略、样式渲染方式、脚本解释器的性能、页面大小、页面组件的数量、页面组件缓存状况、页面组件域名分布以及域名 DNS 解析等，并且其中一些因素随着各厂商浏览器版本的不同而略有变化。这部分内容我们在后续章节中也会适当提到。

可见，一个页面包含了若干个请求，每个请求都或多或少地涉及以上这些过程，假如有一处关键环节稍加拖延，整体的速度便可想而知。

现在，如果有用户向你抱怨在打开站点首页的时候等待了很久，你知道究竟慢在哪里了吗？

1.2 瓶颈在哪里

相信你一定知道赤壁之战，这是中国历史上一场著名的以少胜多的战役，东吴的任务是击退曹操的进攻，要完成这项任务，可谓“万事俱备，只欠东风”，这时东风便是决胜的瓶颈，所以很多系统论研究专家将其称为“东风效应”，也就是社会心理学里讲的“瓶颈效应”。

之所以称它为瓶颈，是因为尽管东吴做了很多的战前准备，包括蒋干中计导致曹操错杀蔡瑁和张允、诸葛亮草船借箭、东吴苦练水军等，但是仅靠这些仍无法获得最终胜利，还需要最后的东南风才能一锤定音，完成火烧曹军战船的计划。不过之前的准备工作都是胜利的子因素，而东南风这个关键因素最终和其他子因素一起相互作用，将整个战斗的杀伤力无限放大。

曹操运气不好，遇上东南风，倒了大霉，曹军战船一片火海，这时候东吴需要派出勇猛的陆军部队登岸攻下曹营，可是东吴向来精通水战，几乎没有强大的陆战部队，只有老将黄盖，这如何与曹操的精英骑兵抗衡呢？这个时候决胜的关键因素变成了刘备的盟军支援，五虎上将各个威猛无比，身怀必杀绝技，此时正是上岸一显身手的好机会，他们不费吹灰之力就将曹军打得落花流水，试想如果没有刘备的支援，赤壁一战胜败可能就扑朔迷离了。

可见，系统性能的瓶颈，是指影响性能的关键因素，这个关键因素随着系统的运行又会发生不断的变化或迁移，比如由于站点用户组成结构的多样性和习惯的差异，导致在不同时段系统的瓶颈各不相同，又如站点在数据存储量或浏览量增长到不同级别时，系统瓶颈也会发生迁移。一旦找到真正影响系统性能的主要因素，也就是性能瓶颈，就要坚决对其进行调整或优化，因为你不得不这么做。

提示：

中医是一门关于生命的哲学，也是中国人智慧的结晶，它的光芒在于独到的思辨能力和系统性的分析方法，它认为世间万物都在不停地变化，并赋予它们阴阳状态，包括天地、季节、天气、心理、生理等，而患者的病理也在随之变化，所以，中医会对同一位患者在不同季节进行不同的诊断，找到不同的病因。

同时，在这些关键因素的背后，也存在很多不能忽略的子因素，构成了性能优化的“长尾效应”，也就是说如果你对某个子因素背后的问题进行优化，可能会带来性能上的少许提升，也许不被察觉，但是多个子因素的优化结果也许会叠加在一起，带来性能上可观的提升。对于诸多子因素的优化，需要稍加谨慎，花点时间考虑这种优化是否值得，以及是否会带来潜在的副作用，还有其他依赖的非技术因素。

然而，不论是关键因素还是子因素，它们的背后都是影响系统性能的问题所在，问题本身并不涉及关键性，只有在不同的系统和应用场景下，才会显示出其是否关键。

本章的其余部分将先列出一些我们经常遇到的问题，并简单介绍我们常用的优化方案，至于这些问题在什么时候是否关键，它们的本质是什么，以及如何调整或优化，在后续章节中我们将结合具体场景来详细探讨包括这些在内的更多主题，这也是本书贯穿始终的线索。

1.3 增加带宽

当 Web 站点的网页或组件的下载速度变慢时，一些架构师可能想到的最省事的办法就是增加服务器带宽，因为他们认为是服务器带宽不够用了，对于一些以提供下载服务为主的站点来说也许是这样的，但是对于其他服务的站点，你知道站点当前究竟使用了多少带宽吗？这些带宽都用到哪里了呢？如何计算站点现在和可预见未来使用的带宽？带宽增加后下载速度就可以加快吗？使用独享带宽和共享带宽的本质区别是什么？如何节省带宽？还有，你可能会忍无可忍地问，究竟什么是带宽？

对于带宽的概念，如果你没有仔细阅读计算机网络教材中的描述，我敢肯定你一定是完全凭借自己的理解来认识它的，因为这个词实在是太有创意了，也实在太容易从字面理解了，