HZ BOOKS

Internet 新技术丛书

# Perl Web 开发技术

## Perl for the Web

您用什么语言来开发Web?
调查显示——超过60%的开发人员会回答: Perl!

(美) Chris Radcliff 著
王爱国 周丽萍 等译

机械工业出版社
China Machine Press
New Riders

Internet新技术丛书

# Perl Web开发技术

(美) Chris Radcliff 著

王爱国 周丽萍 等译

前导工作室 审校

机械工业出版社
China Machine Press

本书重点介绍了用Perl进行Web开发时可能遇到的问题、现在的解决方案和未来的发展趋势。

本书主要内容分为三个部分，第一部分具体讨论了Web开发时可能遇到的、应该注意但却可能被忽视的一些问题，这些问题在用Perl语言或其他语言进行Web开发时普遍存在；第二部分介绍了用Perl开发Web应用程序的优势，并提出了对问题的解决方案，包括基本原理、实现工具和存在的优缺点等，并具体针对Web应用程序的性能问题提出不同的解决方案，这些方案都充分体现了利用Perl的优势；第三部分介绍了面向未来的解决方案，具体介绍了Perl的未来、Web开发的发展趋势和一些示例。

本书编排独特，阅读方便，针对性极强，是一本不可多得的好书。本书是用Perl进行Web程序设计和提高Web应用程序性能的很好的参考资料。对利用其他语言进行Web程序设计和开发的技术和管理人员也有一定的指导作用。

Chris Radcliff: Perl for the Web (ISBN 0-7357-1114-3).

Authorized translation from the English language edition published by New Riders Publishing, an imprint of Macmillan Computer Publishing U.S.A.



**本书版权登记号：图字：01-2001-4282**

**图书在版编目(CIP)数据**

Perl Web开发技术/（美）拉德克利夫（Radcliff, C.）著；王爱国等译. -北京：机械工业出版社，2002.2

（Internet新技术丛书）

书名原文：Perl for the Web

ISBN 7-111-09779-3

Ⅰ.P… Ⅱ.①拉… ②王… Ⅲ.Perl语言-程序设计 Ⅳ.TP312

中国版本图书馆CIP数据核字（2002）第006279号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）

责任编辑：翟静华

北京第二外国语学院印刷厂印刷 · 新华书店北京发行所发行

2002年2月第1版第1次印刷

787mm × 1092mm 1/16 · 18.25印张

印数：0 001- 4 000册

定价：35.00元

# 译 者 序

Perl是Practical Extraction and Reporting Language的缩写。SUN微系统公司（http://soldc.sun.com）曾面向广大开发人员进行过一个在线调查“您使用什么语言来开发基于Web的客户端应用程序？”，结果显示：Perl（62.1%）、JSP(12.5%)、PHP(8.7%)、C/C++(7.1%)、ASP(5.7%)、其他(3.7%)。不管这个结果是否准确，它确实说明了Perl语言在Web开发上的流行性。

Perl 编程语言的设计者 Larry Wall把Perl称为后现代语言，它的目的不是要推出一种新的风潮，而是集各种语言的优势，解决实际问题。Perl强大的功能、简洁的语法，使它受到了广大程序员的特别喜爱，从而成为一种非常流行的工具。使用Perl开发出的应用程序数以万计，公共代码的Perl库函数覆盖了计算机的各个领域，成为其他语言所无法比拟的一个资源，这些库函数都可以在www.cpan.org找到。

从语言功能与性能来看，Perl与任何一种现代语言相比都毫不逊色，这主要表现在：快速的开发过程、极强的可移植性、极丰富的语言功能和文本处理能力、高超的性能、对OOP的强大支持、与C/C++的接口和非常方便、极为丰富的函数库等等。

Perl很优秀、实用，并且已经被广泛应用在各个方面，特别是在互联网上。上述SUN微系统公司的调查就很好地证明了这一点。

本书是关于提高Web应用程序性能的很好的参考资料。我们在拿到这本书之后，就被书中的内容吸引住了。我们原来在Web性能改善方面比较模糊的地方，在本书中都找到了答案。我们对这本书总的感觉就是实用、易读。

本书独特的安排形式——从基本到高级，从现在到未来，从总体上挖掘性能问题的根源，并有针对性地提出各种解决方案。本书将复杂的主题分解为容易处理的各个小节，这种组织形式有助于读者快速找到自己所需要的部分。

我们相信，本书不仅对Perl面向Web应用的入门者有帮助，对于那些熟悉Perl和Web编程的人们，当他们遇到困难时，翻阅本书，一定会体会到“山重水尽疑无路，柳暗花明又一村”的感觉。

全书由王爱国、周丽萍、李为、王大用、张非非、陈立文、从亚丽、崔玉宝、张中、何镜、高松、叶小松、李林、张巧丽、蒋华、施平安、张勇、鲁金贵等进行翻译，前导工作室全体工作人员共同完成了本书的翻译，录排，校对等工作。本书最后由王爱国、周丽萍统稿。由于时间仓促，且译者的水平有限，在翻译过程中难免会出现一些错误，请读者批评指正。

如果您在阅读中碰到了什么问题，请同前导工作室联系：reader_service@263.net。我们会尽力解决您的问题。

2001年10月

# 前　言

设计一个Web应用程序不是进行脑部手术，确实，它没有那么复杂，它的使用与单一用户或企业级应用程序比起来也是轻量级别的。使用Perl编写应用程序使工作更容易。尽管Web应用程序设计很简单，但其中包含的一些基本思想被大多数应用程序设计者完全忽略了。这些思想不是很复杂，本书真正重要的部分可以用一两页总结出来。但这些原则仍被如此多的人如此经常地忽视的主要原因是它们很新，以前还没有把它们写下来。

我应该在此更正，这些原则其实早已有人写出来，只是比较分散。我初次知道其中的一些原则是在阅读文档时偶然发现的。其他的原则是我在让一个不是极度缓慢的Web应用程序工作起来的反复试验中，以及点点滴滴地改善其性能时学到的。这些原则中的一些完全是我独自创造的，我长期以来一直把它们作为工作原则，它们已与本书的其他部分融为一体。

那么，既然基本东西可以用一页纸概括，为什么要用一本书的篇幅呢？一个原因是您不可能只拿起一页的小册子然后用它的原则来开发一个百万美元的Web应用程序。另一个原因是您不可能拿一页的小册子让您的老板相信这就是开发应用程序方式的足够的理由。当讨论的技术是Perl时，这个问题尤其尖锐，因为没有亿万美元的公司作广告宣传它。提出一个Perl解决方案可能是一场艰苦的战斗，因此，拥有基于实践经验的一个相关的、完全的参考是有帮助的。

写作本书的主要原因是一页纸不能详细覆盖所有的重要思想。它可能包括一些基本的思想，但它缺少提出那些思想的重要原因。它不会包括使CGI缓慢的体系结构模型或使基于模板的HTML成为必须的业务处理过程。它不会谈到15件小事加起来使基于Web的数据库访问速度变慢，并且它也不会提及为什么它不总是最有效的原因。最重要的是，它不会用不同的例子来解释它的观点，而任何一个例子都能使思想更进一步被领会。

## 读者对象

如果您是一个Web站点管理员、一个Web网站项目的带头人、一个CTO、一个数据库管理员或您只是一个寻求更好的方法的Perl黑客，这本书就是您想要的。这本书的读者对象包括要改善其Web应用程序性能的程序员，试图评估Web技术对一个项目可用性的管理者和希望为其Web页面添加程序处理而又不想陷入CGI深潭的HTML设计者。这本书也对那些试图要说服别人采用比他们现有策略更好的人都有用。

我衷心地希望那些已经厌倦的人来阅读本书，有的人可能厌倦了CGI程序的深不可测的性能，另一些人可能厌倦了Cold Fusion、ASP或Java servlets的较差的可伸缩性，还有更多的读者可能厌倦了编写面向对象的mod_perl程序的需要。管理者可能厌烦了程序员的喋喋不休地对性能测试、更好的硬件和更Perl化的要求。

对Perl不熟悉的技术性和非技术性的读者在第1～7章可以发现最有价值的决策见识。我已经尝试把本书的其余部分写得尽可能地清楚，以适合非程序员来理解，并且每章的引言和小结部

分应该所有人都能掌握。

## 阅读本书的先决条件

为了能毫无阻碍地阅读本书，您应该对Perl编程和Web设计有一定的了解。具有用Perl编写CGI应用程序的经验的任何程序员应该具有利用本书原则优势的必须的背景。没有Web经验的Perl程序员可能需要对HTML进行熟悉。没有编程经验的HTML设计者可能在开始处理示例之前对Perl的知识要进行补充。

另外，如果您是PHP和ebXML（在此插入您最喜欢的语言和XML变种）的高手，并且想转到Perl，那么您会发现所有事都相当熟悉。如果您想用Perl和XML做一番工作，您不需要从本质上了解HTML的所有东西，从XML观点来看，示例是很好的选择，并且第三部分“面向未来的解决方案”正是您所需要的。

## 本书的组织

本书分为三个部分，包括存在的问题、解决方案和未来Web应用程序开发的方向。第一部分“存在的问题”定义了Web应用程序设计时遇到的一些问题，它是为不了解Perl的技术型读者设计的。对程序员，第一部分（包括第1～6章）最好用于确定您所遇到的问题或可能遇到的问题。对决策者，它提供了对Web应用程序设计者从开始配置到多年实现过程中遇到的问题的概览。

第二部分“解决方案”（包括第7～15章），主要为Perl程序员在一个Web应用程序设定中解决问题提供帮助。这些可能是您正在寻找的解决方案：改善性能，创建快速原型，添加更多的模块性，取悦网站设计者和测试网站性能。如果您认为您已经覆盖了所有这些基本情况，阅读第一部分，来重新发现问题。

第三部分“面向未来的解决方案”（包括第16～21章），是您在解决前两部分问题之后的可能情况。XML、SOAP、WAP、WML、DOM、XHTML、UDDI和WSDL都在此出现。群集、反编译和多重记录集合都非常容易让人混淆，所以要仔细研究其原理。

# 目　录

## 第二部分 解 决 方 案

## 第三部分 面向未来的解决方案

# 第一部分　存在的问题

- 不可预期的网络流量问题
- 不理想的预算和计划
- 网站设计和应用程序设计
- 原型和实际网站
- 基于体系结构的性能损失
- 经常忽视的问题

## 第1章　不可预期的网络流量问题

在Web上，流量问题既可能是一个应用程序的救世主也可能是其终结者。没有比缺乏流量更能致一个Web应用程序于死地的，因为这暗示着程序缺乏吸引力，并且最终的销售量会很低。反过来说，网络流量的增加意味着对有限的硬件和软件资源的需求更大。

### 1.1　过时的网站

每一个网站都希望能永世常青——永世不老、永不过时或不会因任何原因而失败。Yahoo!就是这样的站点之一，它的核心内容自建立之日起就保持相同，在投入使用以后，网站的设计改变也很小，它是如此地可靠，以至许多人用它来检验自己的Internet连接是否失败，换句话说，如果您不能到达Yahoo!，您不能到达任何站点。

当然，不是每个站点都是Yahoo!，也不是每个网站都是可访问的、快速的和可靠的。当站点的反应缓慢，网页是间歇式地响应或站点不可访问时，对正在访问站点的人和正在运行站点的人都可能会造成很多问题。

我在冲浪过程中搜集了一些过时的Web网站的例子。大多数网站并不是由于特别的原因变得不可访问的，而是由于网络的流量超过了设计者的预期目标。这些问题很容易解决，我已经设计过一些站点，它们运行的时间超过了预期寿命好多年，因此当它们失败时，并不令人十分震惊。

#### 1.1.1　macadamia的故事

节日是爱的日子、欢乐的日子和消费的日子，尽管我做过许多尝试，但还是没能跳出这个套路。当然，作为21世纪的人，我喜欢在线购买商品。这意味着可能只能买到亲人不喜欢的礼物，并且这些礼物价格很高，但不同的是这些礼物是我在戴着拳击手套，听着Muppets的圣歌，

喝着饮料，玩着挖雷游戏的同时在Web网站上买到的。

但不幸的是，1999年的节日并不像我所预期的那样方便。那是我第一个完全在线购买礼物的节日，我非常高兴在一下午就能买到所有的礼品，而此时我的邻居正在当地购物中心的一次大降价活动中进行肉搏战呢。事实上，我对自己在线购买所有东西的能力太乐观了，直到最后一刻我才下订单，并且我听说大多数站点除非当天下订单，否则并不能保证准时送货。当然，大多数站点并无此声明，但我想还是保险一些好。

我最后要买的礼物，也是最重要的礼物，是送给我母亲的。她是一个可爱的老太太，她特别喜欢macadamia坚果。所以，我的目标很明确，找到一个提供各种包装精良的macadamia坚果的站点，我的购物就大功告成了。作为一个挑剔的购物者，我计划访问一系列网站以确保买到的礼品物美价廉。您知道，多浏览窗口提供的购物体会是购物中心无法比拟的。我的第一个目标是Harry&David，这是一个我经常光顾的公司，无论是通过邮寄订货还是到它的店里。我已经注意到他们有一个网站，所以，这看起来是搜寻macadamia坚果的好去处。

Harry&David站点的布局很好，但我打开首页时就注意到一个性能问题。我已习惯于等待网页，乐于给网站提供悬念的机会，因为我知道站点建设的内幕；我也乐于等待那些macadamia坚果，因为在网页加载过程中，还有其他的浏览窗口和站点可以访问。不幸的是，这次我不得不为每个网页等更长的时间，我想，这个站点的访问人数一定很多。最后，页面加载超时，但我还是坚持重复加载页面直到有东西出现，就在我终于有机会看到我认为是最好的macadamia商品时，网站返回一个Asp应用程序错误，拒绝进一步显示页面，我被拒之门外了，看来不可能从Harry&David站点买到macadamia坚果了。

我下决心要继续寻找。当然，这里还有许多站点提供我想购买的macadamia坚果，并且站点的稳定性能满足我实际下订单的需求。我一个网站一个网站地浏览，无论大的还是小的，但它们都反应缓慢，或遭到破坏，有的甚至既慢又遭到破坏。一些站点在我试图下订单时超时，其他站点在我搜索“macadamia”时给出应用程序错误并就此退出，而我连坚果有没有都不知道。到了我该决定是继续下去还是放弃并加入到购物狂潮中去的时候了。但我不能放弃，因为我相信总会有一个地方会卖给我一些macadamia坚果。

最后我采取了一个确实有点疯狂的举动，我找到了macadamia坚果制造商Mauna Loa的站点，它的总部在夏威夷，这一站点不是为电子商务而设计的，我根本不能买到实际食品，于是，我转到站点的样品部分，在那里可以货运到其他没有macadamia售货的地方，我提交了我的信用卡号码和我母亲的地址。没有礼品包装，没有留言，盒子里装的只是坚果。简而言之，它看起来就像来自超级市场，没有货运日期的保证，可能是通过美国邮政，但它奏效了，我可以在线定货，那才是最重要的。我让母亲知道macadamia已在路上，让她知道我很爱她，她就已经很高兴了。

后来，我在节日后打电话问母亲macadamia坚果是否已收到，答案是否定的。可能是由于邮寄服务的疏忽或是Maunu L oa的运输错误，也可能是订单本身被疏忽了——一个在加利福尼亚的人从夏威夷订本地就有销售的坚果到威斯康辛，难道这不令人奇怪吗？不管什么原因，坚果没有收到，我有时想象它们正在全国漫游找寻我的母亲，但这永不会有结果。

**案例研究：WFMY**

Chris Ellenburg知道为增加的网络流量做准备的重要性。Ellenburg是WFMY新媒体的主任，并且是该网站的Web管理员，这个网站是在1999年开始运行的。WFMY是电视行业中第一个提供Web站点的电视台，它获得了巨大的成功，而站点的流量也因此超过了站点所能处理的数量。

Ellenburg最初使用Perl CGI开发WFMY Web站点，并连接到提供动态内容的MySQL数据库上。新闻报道、天气预报、节目清单和文档内容都由Web应用程序处理，因为站点上的大多数页面需要提供即时信息，这些信息使网站迅速在地区和全国的访问者中取得成功，但这种成功也迅速导致崩溃。

在WFMY网站公开不久，Floyd飓风（可能是北卡罗林纳州历史上最猛烈的飓风）引起了全国的注意，站点收到大量的关于飓风信息的请求。WFMY是这个地区权威性的Internet新闻信息源，因此，网站受到请求的连续冲击——那个月的站点访问记录超过三百万。流量冲击超过了站点的CGI程序的处理能力，正如Ellenburg很快发现的，站点的响应速度变得很慢，需要20秒钟来返回每个页面。最终，这个站点停止返回任何页面，并且站点Web服务器本身由于负载过重而每10分钟宕机一次。在飓风过去之前没有时间来改善性能或阻止流量。而站点的使用量在这之后降到一百万次每个月，这是服务器可以处理的。

Ellenburg意识到要采取措施防止同样的事情再次发生。所以，他权衡解决网站性能问题的各种解决方案。最初考虑将服务器升级，这包括在其原始的500M的RAM基础上再增加RAM或使用负载平衡硬件聚合一组服务器。这两种选择对站点都太昂贵了，但是，每种选择都可以带来速度的很大提高。Ellenburg也考虑在整个站点使用静态HTML文件来减少CGI处理的开销，但即时信息是站点的核心，因而这也不可行。尽管将引起Web站点性能下降，但动态页面和Web应用程序还是无法舍弃的。

很清楚，问题的解决方案要对站点的Perl处理程序的性能进行改善。Ellenburg找到mod_perl和VelociGen，这两者都声称通过缓存和预编译基于Perl的应用程序，从而提供对CGI的性能改善（mod_perl、VelociGen和其他改善性能的工具的详细信息参见第10章“Perl持续性工具”）。

这两个包都名副其实，比较原始CGI与使用嵌入Perl的持续性程序的测试显示持续性应用程序比CGI版本的程序的快10倍（参见第12章“减少开发时间的环境”）。在测试过程中，Ellenburg也发现他欣赏直接嵌入Perl到动态HTML页面的容易性。

他们使用嵌入Perl程序开发了一个新的WFMY网站，它在2000年10月开始运行，并得到一致认可。网站访问者立即发现站点的响应速度有明显改观，更重要的是，Ellenburg的老板也发现了。网站再次记录了每月三百万的点击率，这次没有什么问题。不再受到站点响应的限制，流量在2000年12月上升到七百六十万，其中单纯页面点击就是二百万。Ellenburg估计网站现在的配置可以很容易地处理每月二千万的点击率。当下一次飓风来临时，无论它是数字的还是气象的，Chris Ellenburg都成竹在胸了。

### 1.1.2 评估宕机的费用

WFMY没有因其站点响应速度慢而失去任何业务，但电子商务网站一定会受到响应速度慢和不可访问的影响。在电子商务站点，反应慢或不可访问的费用可以通过估计在此期间收入的损失计算出来，例如，如果Amazon.com在2000年的收入是18亿美元，则每小时的宕机费用为280 000美元，这意味着如果网站的可用性从99.9%提高到99.99%，每天不可用的时间从90秒减到9秒，则单从收入上讲每天就增加6 300美元，或约190 000美元每个月，这还没有包括站点职工的生产能力的提高、网站延误的改进或其他内部费用。

以上数字也能说明提高性能所降低的费用。即使一个服务器时间上提高到99.9%，仍可能慢得打击潜在客户浏览本站的积极性。一个速度慢的站点会使客户大幅度减少，如WFMY所证明的；它们的站点使用率在网络速度慢时下降而在速度快时明显增加。在这种情况下，估算损失可能不太可靠，却更具挑战性。如果Amazon失去总客户数的25%，那它在9个月中损失4.5亿美元；如果它像WFMY那样改善性能，从而顾客增长600%，它的收入则会增加110亿美元（钱对一个努力获利的公司总不是一件坏事）。这种扩张的想法并不是革命性的和不可检验的，正如一个传统的商店肯定要扩充规模而不会把顾客拒之门外一样。

当然，这些数字都是估计的，不同企业之间也不相同，但原理总是正确的：Web网站性能的提高可以赢得访问者和收入，并且在评估性能提高和网站结构升级的费用时，还应该权衡升级和不升级的机会损失。

### 1.1.3 提前准备的费用

为一个网站的流量增加做准备的费用可能是巨大的，但如果做准备的实际费用已知，并在升级开始之前进行计划，就可以避免做成本的浪费。知道升级的潜在费用还有另外的好处，一个好的体系结构方案可以提高性能以及降低成本，这对追求最大收益的管理者更具吸引力。

性能检查的费用可分为四类：

- 调查性能问题和确定解决方案的时间。
- 硬件升级费用。
- 软件升级费用。
- 培训开发和管理新系统员工的费用。

通常，只考虑新增硬件和软件的费用，并且假定已经计算调研和开发的时间，而且员工费用对所有方案都相同。然而，这四个领域中每个选择都会影响到其他领域的费用。

对所有方面的总体评估能更好地体现一个方案的真实费用。例如，通常选Windows作为服务器端操作系统，因为培训Windows管理员比培训Linux管理员容易，且雇佣费少。不幸的是，处理与Linux服务器同等的负载时需要更多的Windows服务器费用，这和增加宕机时间和重新配置需要综合起来考虑，就需要雇佣更多的管理员，因此附加的雇佣费会与个人工资结余相低消。

## 1.2 突然的流量增加

突然的流量增加可能出于多种原因，其中最有可能的是Web规模在增加，并且一些流行站点

间互相协调。一个站点可能因为人们对新的营销或网站的产品、服务或信息的普遍兴趣而导致网站流量剧增；网站也可能与另一个受欢迎站点或服务链接而变得流行起来，这导致在很少甚至没有任何警告的情况下流量大增。

### 1.2.1 名声和Slashdot效应

Slashdot(http://www.slashdot.org)是一个网站，它合成Web上新兴社区可能感兴趣的新闻和产品评论，网站的座右铭是："为大众提供新闻，揭开事实的真相"，并且从1998年创建以来拥有了一大批的追随者。列举Slashdot的例子是为了说明一个新的想法，或声明一项新的技术的可用性，而不是像晚间新闻那样评论一件新兴事物的状态。Slashdot并不只局限于技术文章，任何能引起新兴一族的兴趣的新的发现、理论、产品、服务或想法都可以提供，并且，每天都有成千上万的Slashdot网民提交帖子以使自己的名字列入最新的大字标题下。

2001年初，Slashdot有几百万读者。许多人每天都访问网站，有些人（像我）每天在编写新兴社区新闻的时候要多次访问网站。正因为如此，数以百万的Slashdot的读者可能在任何时刻对一个故事做出反应，这与提供每周或每日新闻的网站的一整天中的访问量不同。

为了适应这种模式，Slashdot在任意给定时间内把大多数读者的焦点集中在某一篇文章上。网站把一天中有价值的头条新闻列在主页上，并且通常都与网站或讨论的新闻链接。随着头条新闻的不断更新，这些内容被列在与之分开的档案区。因为这些少量链接的快速可用性，许多Slashdot读者可以在很短的时间间隔内点击同一链接，这种在很短时间内引起众多人的兴趣的效应称为Slashdot效应。

Slashdot效应使流量产生巨大的变化，这足以使许多对注意力很慷慨的Web网站瘫痪，因此，后来Slashdot故事的评论中提到：这些网站都是被"Slashdot"掉了。这经常发生在相对不太出名的网站上，因为Slashdot效应产生比它们常规处理能力大得多的流量。

Slashdot效应在Slashdot读者还没有看到链接的站点时，就可能更加扩大化了，实际上，小网站可能误把这作为分布式拒绝服务攻击。如果输入的请求导致网络连接的饱和，就很难辨别合法兴趣和恶意攻击的区别。然而，在大多数合法兴趣的情况下，Web服务器早在网络连接建立之前就已经过载了，这样就很容易分清它们之间的差别。

即使网站的内容或功能在Slashdot规定的焦点之外，也没有网站和主题对Slashdot效应是安全的。因为Slashdot的编辑者有独立的编辑控制权，任何一位编辑都可能有自己的故事要贴上去。

### 1.2.2 其他名声

Web上有许多网站产生与Slashdot效应类似的效应，但这些网站的流量是在较长时间内积累起来的。像Yahoo! News或C | Net这样的网站只需简单地在一篇文章中提一下，就可以给一个以前不太出名的网站带来许多注意力；如果美联社或路透社合成了这篇文章，效果将更为显著。通过这样的合成，文章可能同时出现在许多新闻网站上。Web上的新闻作者也知道利用彼此作为信息源，即使一个没有多少信息量的报道也可以一再作为其他新闻条款的片段被引用。

当一个Web网站提供了一个Web应用程序或工具箱时也可以看到流量的突然增加。如XML