HZ BOOKS
华章科技
PEARSON

由拥有10余年工作经验的资深系统运维工程师撰写，Nagios创始人兼总裁亲自作序推荐，权威性毋庸置疑

深度阐述系统监控的最佳实践和运作原理，全面而系统地讲解Nagios系统监控的各项功能、技术和方法，为构建高效的企业级监控平台提供有效指导

Nagios: Building Enterprise-Grade Monitoring Infrastructures for Systems and Networks

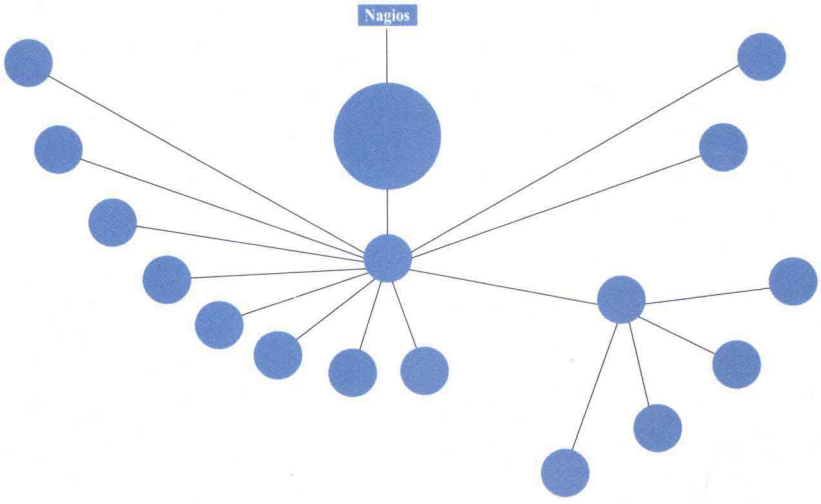
Second Edition

Nagios系统监控实践

（原书第2版）

（美） David Josephsen 著

康锦龙 译



机械工业出版社
China Machine Press

Nagios: Building Enterprise-Grade Monitoring Infrastructures for Systems and Networks

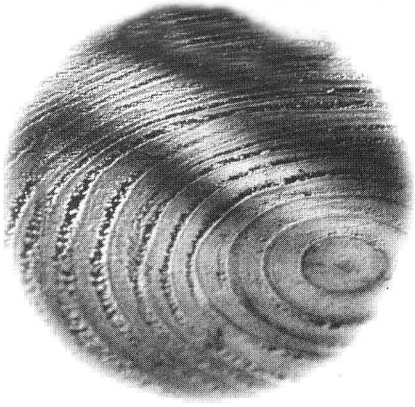
Second Edition

Nagios系统监控实践

（原书第2版）

（美） David Josephsen 著

康锦龙 译



机械工业出版社
China Machine Press

图书在版编目（CIP）数据

Nagios 系统监控实践（原书第 2 版）/（美）约瑟夫森（Josephsen, D.）著；康锦龙译 . —北京：机械工业出版社，2014.1
（华章程序员书库）
书名原文：Nagios: Building Enterprise-Grade Monitoring Infrastructures for Systems and Networks, Second Edition

ISBN 978-7-111-45361-1

Ⅰ . N… Ⅱ . ① 约… ② 康… Ⅲ . 监控系统 – 计算机网络管理 Ⅳ . TP277

中国版本图书馆 CIP 数据核字（2013）第 321118 号

本书是介绍 Nagios 的权威指南。详细讲解了整个监控技术，演示了最佳做法，揭示了常见的错误及其后果，以及如何避免。提供了所有配置和运行方式，并探讨如何编写自定义模块与基于 Nagios 事件代理 API。

本书从实际出发，在开篇就系统运维中的监控提出一系列需求，从而展开对 Nagios 系统的初步介绍（第 1~2 章），随后从实用的角度，全面、详细地讲解了 Nagios 安装、配置的相关内容（第 3~4 章）。通过简化配置、实施监控等工作（第 5~6 章），用大量的示例展示 Nagios 的实际能力。然后，在扩展方面介绍了一些常用的方案（第 7 章），并从原理、案例到最后的 DIY，一步步带领读者进入数据可视化的世界（第 8 章）。此外，还介绍了 Nagios 商业版本——Nagios XI 的功能特色（第 9 章）。最后，介绍 Nagios 事件代理（NEB），并用 C 语言实现完整 NEB 插件（第 10 章），使读者进一步掌握 NEB 的工作机制。

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：肖晓慧
藁城市京瑞印刷有限公司印刷
2014 年 1 月第 1 版第 1 次印刷
186mm × 240 mm • 15.5 印张
标准书号：ISBN 978-7-111-45361-1
定　　价：69.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
客服热线：（010）88378991　88361066　　投稿热线：（010）88379604
购书热线：（010）68326294　88379649　68995259　　读者信箱：hzjsj@hzbook.com

译 者 序

本书是我的第一本译著，有此机会实属机缘巧合。虽然使用 Nagios 只有一年多的时间，但是作为用户，我深感其设计的简洁与高效——没有一丝多余的东西。因为工作的关系，要求对各个领域都有所了解，所以没有仔细地阅读 Nagios 官方文档，但总以为自己了解的内容差不多了。但在翻译本书的过程中，才发现自己想当然了。

作者开篇就从实际出发，提出了几个引人深思的问题，这些问题恐怕是国内外企业在构建监控系统时都会存在的。对于这些问题我个人深有体会，在 Nagios 使用的过程中曾经踩到不少坑，直至年中时遇到的性能问题、分布式部署问题。至于系统构建完成时又遇到的商业产品竞争问题，具体细节我就不详述了。长叹一声，当时要有这本书，就能少走不少弯路。

在本书中，作者介绍了在大量运维监控方面需要了解的知识，但偏重于 Linux——因为 Nagios 是专为 Linux 环境设计的。这可能会让不少读者们感觉不快，但无论从哪一方面来说，Linux 的高效是无可比拟的，可以说，Nagios 与 Linux 都是一种效率的必然。

对于 Nagios 用户来说，或许早已深感 Nagios 的配置是一种简单但重复性很强的工作。在本书中，作者从实际出发，引入了“框架”的概念，从而简化了配置工作。不仅如此，在扩展、数据可视化以及最后的 NEB 方面，作者都通过代码或者案例进行展示，方便各位读者理解概念，并起到抛砖引玉的作用。

无论企业的规模大小，Nagios 都是不可多得的开源监控系统，估计这也是 Nagios 作者将开源版本命名为 Nagios Core 的原因吧，期望用户能够围绕 Nagios Core，构建完整的监控平台。

以上纯属个人之见。

感谢机械工业出版社华章公司的各位编辑，感谢各位在审校环节的辛勤付出。最后，感谢 GF 对我长期的支持。

序　言

常听人说 Nagios 非常“灵活”，我想他们是指 Nagios 很容易扩展，但这不是重点。Nagios 在设计中所蕴含的能力不是源自于其扩展性，而是它坚持自身能够被扩展。无法否认，两者区别虽然不大，但这才是最重要的一点。很多软件通过扩展从而完成新的工作，但是只有很少一部分软件，其自身不具有任何功能，除非用户对其进行扩充，而这些软件完全是由于这个原因：内在的需求是用户能够进行定制以满足自己的需要。因此，Nagios 一直是作为用户贡献的合成体而存在的——工程师和管理员用来解决自己面临的问题，并将解决的办法分享出来。没有两个部署是一模一样的，这就是设计上的诉求。

这些年里，从我创建 Nagios 开始，它成长的空间和范围已经远远地超出了我的想象，在全球已经有超过 100 万的用户。从财富 500 强企业集团到艺术科学研究实验室，Nagios Core 在各地落户。Nagios 的用户社区是最健康、最积极贡献的开源社区之一，该社区目前已经发布了近 4000 个插件、附件以及扩展，其中很多东西的复杂性都足够编写一本书来进行说明了。除了社区规模如此之大、多样化、十分活跃之外，我们每年都会举办 Nagios 全球大会，会有贡献者、用户以及学者参加，分享自己的奇思妙想、学习提示以及小技巧，还可以了解 Nagios 项目未来发展的相关信息。

同样，公司内部也有一个繁荣的社区负责 Nagios 的扩展和支持工作。我于 2007 年加入，创立了 Nagios Enterprise。我们的旗舰产品，Nagios XI，不仅迈出了革命性的一步，也（应该是）完全反向兼容于 Nagios Core。XI 包含了扩展即设计（extend-by-design）出身的 Core，同时保留了 Core 的功能和扩展性，并扩充了它的易用性和用户可用性。

在 Nagios 享受着成功喜悦的同时，我需要首先承认，灵活性是要付出代价的。过度灵活会让新手或经验丰富的管理员难以构建以及部署成功的监控解决方案，他们面临的挑战不是有了电脑就可以解决的。幸好，David 这位少有的技术作家，能够将如此复杂的主题写得通俗易懂。无论你是一个对网络、系统、IT 监控知之甚少的新手，还是经验丰富的 Nagios 管理员，本书对你都会有所帮助。

——Nagios 创始人兼总裁，Ethan Galstad

前　言

这是一本关于如何应对不可信设备的书籍，所有设备实际上都不太值得人类信任，但是它们和我们的幸福却息息相关。我用不着讲述一系列恐怖故事来向读者说明流行的计算机系统在故障发生时怎样。如果你拿起了这本书，说明你已经意识到了这些问题：层层叠叠的库相互依赖着，在抽象、脚本小子（script kiddy）[⊖]、病毒、DDos 攻击、硬件故障、终端用户错误、后门、飓风等抽象概念中隐藏着无数的问题。无论根本原因是恶意入侵还是意外发生，你的系统将会出故障，当它们出故障时，能将你从宕机中挽救回来的只有两个办法：冗余和监控系统。

要先选对方向

从概念上来说，监控系统是很简单的：一个外部系统或一个系统群，主要工作就是监视其他系统是否出现问题。比如，监控系统会定期连接某台 Web 服务器以确保它能正常响应，如果出现问题则给管理员发送通知。虽然这听起来简单，但是现在的监控系统已经成为昂贵、复杂的软件系统，其中有很多系统的 Agent（代理）大小已经超过了 500MB，拥有专用的脚本语言，现货标价超过 6 万美元。

如果一个监控系统能够正确地实施部署，它会成为你最好的伙伴。它能在小故障演化成危机前通知管理员，并帮助架构师弄清对应于互操作性（interoperability）异常问题的模式。一个优秀的监控系统能够帮助负责安全的同事将所关注的事件关联起来，为网络运维中心员工展示带宽瓶颈所在，并从业务依赖的关键系统中提取数据，以便为管理层提供他们急需的高层可视化信息。一个优秀的监控系统能帮助用户支撑起服务级别协议（Service Level Agreement，SLA），甚至能在夜晚帮助用户按照预定步骤解决问题，而不用打扰任何人。优秀的监控系统能够帮助用户节约经费，在复杂环境下保持系统的稳定，并使所有人都能安心。

如果实施得不好，监控系统就会带来严重的破坏。拙劣的监控系统会在整晚如同狼那样嚎叫，不

⊖ 脚本小子通常都是一些自发的、不太熟练的 Cracker，他们使用网上下载的信息、软件或脚本对目标站点进行破坏。——译者注

会让任何人睡好觉，从而导致没有人再会关注它。它会在用户安全方面的基础设施上安装后门，从其他项目中榨取时间和资源，并在健康检测的过程中，占用大量的网络资源拥堵用户的网络连接。拙劣的监控系统如同吸血鬼一般。

遗憾的是，第一次就选对方向，对于监控系统的部署来说，没有想象得那么容易。以我多年的经验，拙劣的监控系统不太可能存活到问题被修复的那一天。拙劣的监控系统成为所有人甚至是被监控系统的重担。在这种情境中，很容易明白为什么大企业和政府会雇佣专职的监控专家，并购买标价为 6 位数的软件，因为他们清楚，第一次就“选对方向”是非常重要的。

小型或中等规模的公司以及大学的环境会比较复杂，有可能比大企业还要复杂，但是他们明显不会那么奢侈，也不会拥有价格昂贵的工具和专业知识。面对他们分布在各地的校园和分支机构，部署一套精心构建的监控平台将会是一个挑战。但是，在过去的 13 年里，在花费了相当多的时间负责监控系统的构建和维护工作后，我想告诉各位读者，不仅“选对方向”是可能的，而且还可以是免费的，只不过需要几分辛勤、一些开源工具以及少许的想象力。

为什么选择 Nagios

在我来看，Nagios 这款系统和网络监控工具，是目前可用的、开源的或其他方面中最棒的一款工具。它模块化的设计、直观的监控方式，使其很容易使用，而且高度可扩展。进一步来说，Nagios 的开源许可证使其免费可用，并很容易扩展以满足用户自身的特殊需求。Nagios 擅长与其他开源工具互操作，而非帮你完成所有的工作，这也是其灵活性所在。如果读者想通过本书寻找一个整体化的软件解决方案，能够通过勾选一系列复选框来解决所有的问题，那我可以很明确地说，本书不适合你。但在放下本书之前，希望你能继续阅读一到两个段落，看我是否能够说服你——那种整体化解决方案不是你真正寻找的。

而现实中，绝大多数商业化产品都搞错了方向，因为它们解决问题的方式是假设所有人都希望采用相同的解决方案。在某种程度上来说，这是事实。拥有大量计算机和网络设备的用户都希望如果某些地方故障时能够收到通知。所以如果读者希望销售监控软件，很明显的方式就是创建一款软件，它要了解如何监控现有计算机软件以及网络设备。但对于那些销售监控软件的人来说，他们认为监控系统是个一站式解决方案，在这场较量中，谁能监控的东西最多，谁将最终胜出。

我使用过的所有大型商业软件似乎都遵从着这一逻辑。与（Google 的）Borg 不同，这些商业软件有条不紊地寻找着新的计算机设备并将必要的监控代码加入解决方案中。更糟糕的情况是，某些公司通过直接收购那些已经知道如何监控大量计算机设备的公司，并将这些公司的代码集成到自己的产品

中。他们很快痴迷于功能，并创建了一份庞大的产品功能列表，上面写满了支持的设备。因为他们有软件工程师，所以售前工程师会来到你的办公室，露出一排整齐洁白的牙齿，笑着对你的经理说："没问题，我们的系统可以监控这个。"

问题就在于监控系统不是一站式解决方案。在它能够解决问题之前需要完成大量的定制工作，监控软件的销售者和设计实施软件的工程师之间的差异就在这里。当用户试图构建一套监控系统时，一款通过点击复选框的方式来进行监控的软件不会是用户想要的，用户真正需要的是可以很方便地监控需要监控的设备。专有解决方案往往关注要监控什么，而忽略如何监控，这使得专有解决方案很难满足实际应用。

比如，使用 Ping 程序。我所用过的所有监控系统都会使用 ICMP 回显请求，或者叫做 Ping，以这种或那种方式来检测主机可用性。如果想控制一套专有监控系统如何使用 Ping，则可能会立刻发现架构上的限制。比如想设置 ICMP 包发送的数量或者想根据包往返时间的毫秒数而非简单的通过 / 失败，来发送通知。在更复杂的环境下，可能必须使用 IPv6 的 Ping，或者在 Ping 之前先进行端口试探（PortNock ⊖）。这个问题从整体、功能上的解决方式是这些改变意味着核心应用程序逻辑的改变，因此，必须实施。

在我使用过的商业化监控应用程序中，如果上述 Ping 的例子可以引申，那它们可能需要在监控系统专用脚本语言中重新实现 Ping 逻辑。或者换句话说，就是不得不完全抛弃内置的 Ping 功能。对用户来说，可能对 Ping 检测细节上的控制不一定有价值，但是如果用户连最基本的 Ping 都无法控制，那么在用户环境中，对其他更重要的检测，用户又有多少能够完全控制呢？他们假设知道用户想如何 Ping 某个设备，至此，游戏结束，他们永远不会再次考虑这个问题。为什么呢？因为 Ping 功能已经在产品功能列表上了。

目前，监控设备要面向层出不穷的设备，而 Nagios 关注的是模块化，Nagios 包含很多插件，即专用监控小程序，为专用设备和服务提供支持。Nagios 不会去增加特性而搞军备竞赛，硬件支持方面是社区驱动的。当社区成员需要监控某个新设备或新服务时，就会有人编写新的插件并发布，速度往往比商业应用程序支持同等服务更快。实际上，Nagios 将永远支持用户所需的一切，而且无须对 Nagios 进行升级。Nagios 还提供了两全其美的方案，当用户需要支持时，有商业化的选择，也有兴旺繁荣、乐于助人的社区通过众多的论坛和邮件列表提供免费支持。

选择 Nagios 作为监控平台意味着监控效果只受你的想象力、技术能力以及管理环境的限制。

⊖ Port Knocking 类似于地下组织秘密接头，具体方式为：发送一定序列的 UDP、TCP 数据包。当运行在主机上的 daemon 程序（守护进程）捕捉到数据包以后，判断序列是否正确，如正确则开启相应的端口，或通知防火墙允许该客户端通过。具体信息参见 www.portknocking.org。——译者注

Nagios 能够到达你想监控的任何目的，而且过程极为简单。尽管 Nagios 能完成商业应用程序所做的一切，或者更多，无须安装笨重不安全的 Agent，但它常常无法与商业监控系统相媲美，因为当分析表格时，Nagios 没有那么多的检测项目。实际上，如果他们统计正确，Nagios 本身不做任何检测，因为从技术上来说，它并不知道如何进行监控，它希望你能告诉它如何监控。“如何”监控这个问题，很难通过一个复选框来回答。

这本书讲些什么

尽管 Nagios 本身很难，但它是无数工具中唯一能够组建世界级开源监控系统的。同时，它的文档最完备，不仅拥有一系列书籍、精湛的在线文档，还有生动翔实的邮件列表。我撰写此书的本意是补充文档中漏掉的部分。这本书不是关于 Nagios 的，而是讲解如何使用 Nagios 构建一整套监控平台，更多讲述的是构建监控平台的过程，而非配置某个监控工具。

在本书中，我会介绍一些常用的配置模板，但是如何配置和安装 Nagios 不是我的重点。我关注的是带领读者构建一套优秀的监控平台，为读者介绍一些能够增强 Nagios 功能并简化配置的协议和工具。读者需要深入了解 Nagios 内部的工作机制，这样才能够根据自身需求对它进行扩展。因为 Nagios 的功能远远超出你的想象，所以在本书中，我会花一些时间展示它的强大能力。最后，我还会介绍一些与 Nagios 关系不大的内容，比如最佳实践、SNMP、时序数据的可视化，以及微软脚本相关技术，如 WMI 和 WSH 等。

最重要的是，我将以不同的方式介绍 Nagios。提前介绍它有效的调度和通知引擎，这样在讨论内部机制的时候就能够简洁明了一些。我将重点介绍插件定制和调度等，使读者尽早了解核心内容，而非将这些重要信息放在很少有人阅读的高级部分中。

尽管本书各章节内容有些许独立性，但是我尽可能使这本书成为重要参考资料，涵盖了一系列重要信息，因此建议读者从头读到尾，略过你已经熟悉的内容。本书文字不是很多，但是信息量很大，甚至在监控方面最有经验的老手也能发现一些有用的至理名言。

本书各章节相互依赖，在介绍更为普遍的监控概念时，我也会随意介绍一下某些 Nagios 特有内容的细节。因为在软件安装前，需要做出很多重要的决定，所以我以第 1 章的内容作为开始。该章会让读者考虑进行监控的动机是因何产生的？如何取得成功，比如如何开展实施工作，涉及哪些人，要避免哪些问题？

第 2 章基于第 1 章的通用设计原则，从零开始介绍 Nagios 的基本原理，会让读者从细节上明白 Nagios 的工作机制，但没有提供很具体的配置指令，不会让你淹没在配置的细枝末节里。这里离配置

透明化还有很长的一段路要走。

在能够配置 Nagios 监控环境之前，我们需要先进行安装。第 3 章将会帮助读者通过源代码或者包管理器安装 Nagios。

第 4 章讲述配置，这是令人畏惧的一章。首次进行 Nagios 配置对大多数人来说没什么乐趣，但是我希望通过自下而上的方式，只记录常用和必需的指令，提供常用的示例，并指出对象之间的引用关系及如何引用，尽可能减少大家的痛苦。

很多用户在初次使用 Nagios 之后就无法离开它[⊖]了，并且厌恶使用其他的工具。但是，如果大家对 Nagios 都有点儿抱怨，那肯定是配置了。第 5 章讲了一些题外话，并提供了一些有效的工具，以简化配置过程。这些工具包含自动发现工具，还有图形用户界面等。

在第 6 章，我们终于做好了准备，去了解系统监控的本质工作，并提供了一些具体的案例：包括一些 Nagios 插件配置语法以解决现实世界的问题。以监视微软 Windows 系统环境作为开始，然后介绍了针对 UNIX 的监视，最后介绍“其他系统”的监控，其中包含了网络设备以及环境传感器。

第 7 章是第 2 版新增的部分。在过去的 5 ~ 6 年中，大规模网络环境下 Nagios 的扩展已经成为 Nagios 系统管理员处理的最有意思的问题。设备虚拟化和符合成本效益的云服务的爆炸性增长，需要掌控大量小型节点组成的大型并发处理架构。该章介绍几个工具和策略能够使你将监控的负载分散，并建立一个稳定的大规模监控平台对数万节点进行监控。

第 8 章讲述了一个我感兴趣的主题：数据可视化。优秀的数据可视化能够解决其他方式无法解决的问题，我很高兴现在能有一些选择，包括那些即将诞生的工具。通过最流行的可视化工具，如 RRDTool、Ganglia 以及 Graphite 等，可以很容易地将每天 Nagios 提供的时序数据绘制出来，该章所讨论的不仅仅是折线图。

第 9 章也是第 2 版中新增的部分包括，描述了 Nagios 最新的商业版本——Nagios XI。创建它的人也是 Nagios 的创始人，Nagios XI 使用了本书所介绍的多种工具构建而成，是真正集成和易用性的杰作，使 Nagios 监控变得如此简单，甚至我母亲都可以使用（当然，我母亲曾为嵌入式 FLIR 系统编写了经优化的交叉编译器，希望各位读者明白我的意思）。

读到最后，读者应该已经清楚规则了，第 10 章就要教各位读者如何打破这些常规了。就我所知，该章是唯一介绍最新的 Nagios 事件代理接口的纸质内容。事件代理是目前 Nagios 中最强大的接口，掌握它带给各位读者的回报是能够独自重写第 2 章，读者能够从根本上改变 Nagios 的运作或扩展它的各个方面，以满足自身的任何需求。该章叙述了事件代理的工作机制，并带领读者构建一个 NEB 模块。

⊖ 我敢说，这些用户对它一见钟情。

这本书适合什么人

如果你是一个系统管理员，负责着一堆 UNIX、Windows 系统和各类网络设备的管理工作，并且需要一个便宜的世界级监控系统，那么这本书正适合你。与你期望的正好相反，监控系统的构建不能掉以轻心。虽然监控系统可能会与环境中所有基于 TCP 的设备进行交互，但是只需要一丁点这方面的知识。不要以为这会给你多少喘息的时间，系统监控工作教会我的东西比我在职业生涯中做的其他工作学会的更多，并且在我来看，不论你了解多少，使用监控系统都会是不断地挑战假设，加深你对事物的了解，不断扩展你的所学。

为了最有效地利用本书，你应当灵活掌握那些经常使用的文本型网络协议，如 SMTP 和 HTTP。尽管它也能与 Windows 服务器交互，但因为 Nagios 的守护进程是运行在 Linux 上的，这使其有很重的 Linux 风格，即基于文本，所以熟悉 Linux 或 POSIX 类的系统是很有好处的。尽管对编程的技能要求不是很严格，但是你最好熟悉一些编程技能。本书中有不少代码，但我尽可能地使其直观易懂。但是第 8 章使用了大量 C 语言，而代码是在 UNIX Shell 或 Perl 中编写的。

在阅读本书的过程中，唯一的要求就是当你了解主题的相关内容时，要长期保持开放的好奇心。如果有什么内容看不明白，别灰心，尝试在在线文档中查找下，或者在邮件列表中提问，甚至给我发邮件咨询，我会尽我所能帮助你。

祝你阅读愉快！

Dave

感谢

我亲爱的妻子 Cynthia，她的耐心、鼓励以及美丽，我爱她。

Ethan Galstad，Nagios 的创始人，是他的积极促成了本书第 2 版的出版。

该项目的技术评审都非常出色——伙计们，谢谢你们！

最后，十分感谢 Prentice Hall 出版社的编辑们！他们可不像《蜘蛛侠》或《古灵侦探》中的编辑。Debra Williams Cauley 和 Kim Boedigheimer 都是勤勉、能干的专业人士。他们极其耐心、乐于助人，我很感谢他们为我耗费的时间和精力。

感谢大家！

关于技术评审

Mark Bainter

Mark Bainter 领导的系统管理员团队为消息系统的客户们提供大容量邮件系统外包监控和管理服务，拥有超过 15 年的系统管理员经验，专门从事系统集成、监控与自动化。他是自学成才的博学家，还是一位爱用长词的老顽固。Mark 最近与他的爱妻和 4 个孩子定居在德克萨斯，业余时间他喜欢读书、做木工以及陪他的妻子移居各地。

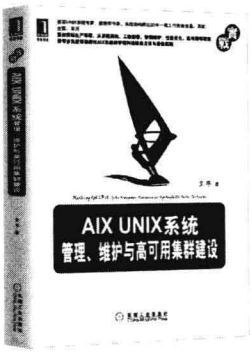
Mike Guthrie

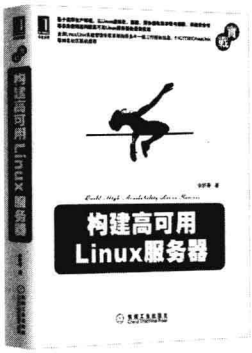
Mike Guthrie 是 Nagios Enterprise 的首席开发工程师，已经为 Nagios Core、Nagios XI 以及 Nagios Fusion 开发了很多新功能和附件。Mike 使用 PHP 完成大多数的开发工作，特别喜欢前端 Web 开发和数据可视化的相关工作。非工作时间里，喜欢和家人待在一起、外出游玩或修缮他的房屋。

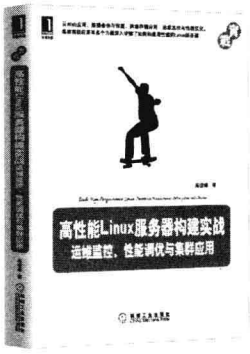
Mathias Kettner

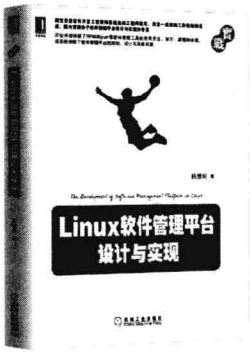
Mathias Kettner 是 Check_MK、MK Livestatus 以及 Nagios 其他附件的作者。他负责德国慕尼黑一家迅速成长的公司运营，该公司专门负责为基于 Nagios 的监控系统提供专业支持和软件开发。

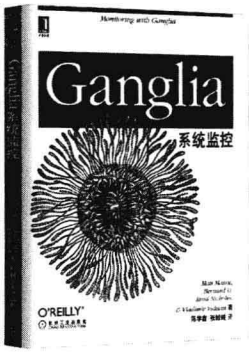
推荐阅读


AIX UNIX系统
管理、维护与高可用集群建设


构建高可用Linux服务器


高性能Linux服务器构建实战
运维监控、性能调优与集群应用


Linux软件管理平台
设计与实现


Ganglia
系统监控
O'REILLY


Puppet实战

目　录