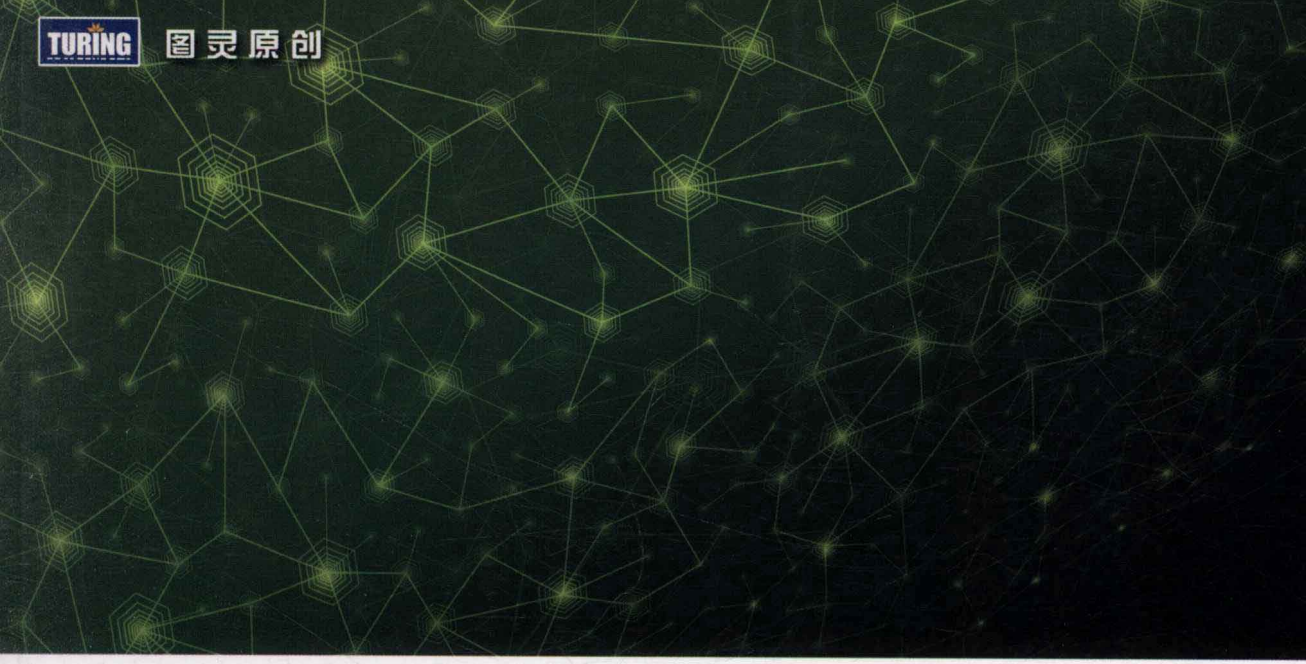
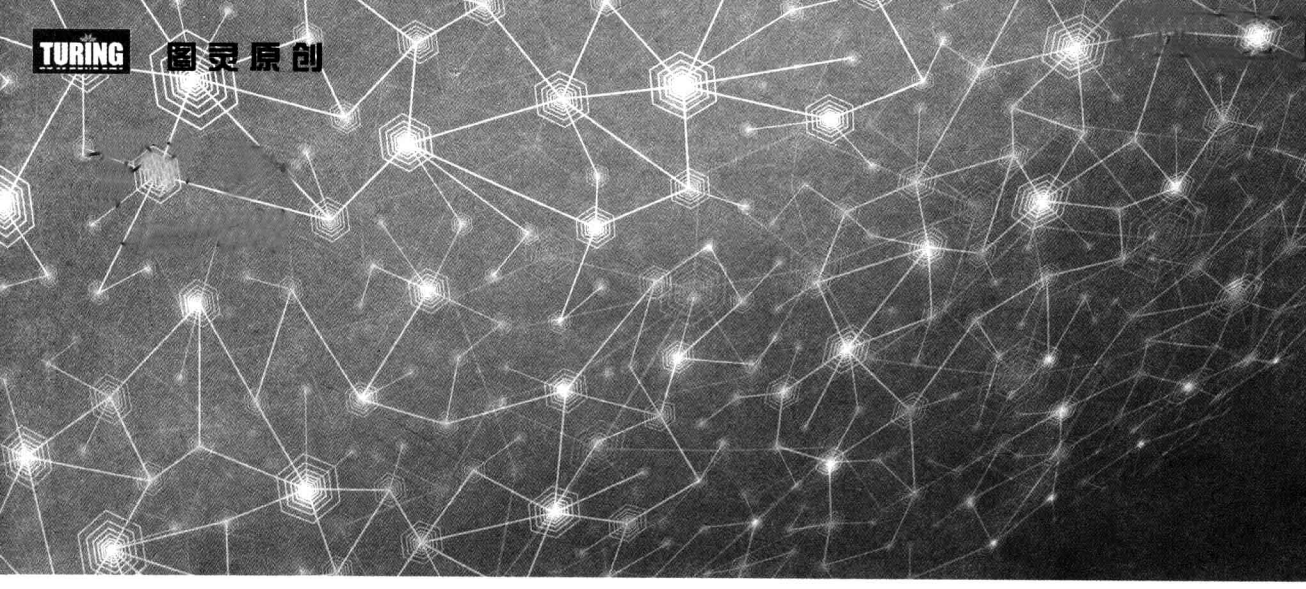


深入浅出 node.js

朴灵 编著

人民邮电出版社
POSTS & TELECOM PRESS



深入浅出 Node.js

朴灵 编著

人民邮电出版社
北京

图书在版编目（CIP）数据

深入浅出Node.js / 朴灵编著. -- 北京 : 人民邮电出版社, 2013.12
（图灵原创）
ISBN 978-7-115-33550-0

Ⅰ. ①深… Ⅱ. ①朴… Ⅲ. ①JAVA语言－程序设计 Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2013)第258737号

内 容 提 要

本书从不同的视角介绍了 Node 内在的特点和结构。由首章 Node 介绍为索引，涉及 Node 的各个方面，主要内容包含模块机制的揭示、异步 I/O 实现原理的展现、异步编程的探讨、内存控制的介绍、二进制数据 Buffer 的细节、Node 中的网络编程基础、Node 中的 Web 开发、进程间的消息传递、Node 测试以及通过 Node 构建产品需要的注意事项。最后的附录介绍了 Node 的安装、调试、编码规范和 NPM 仓库等事宜。

本书适合想深入了解 Node 的人员阅读。

◆ 编　　著　朴　灵
责任编辑　王军花
执行编辑　董苗苗
责任印制　焦志炜

◆ 人民邮电出版社出版发行　　北京市丰台区成寿寺路11号
邮编　100164　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
三河市海波印务有限公司印刷

◆ 开本：800×1000　1/16
印张：21.75
字数：514千字　　2013年12月第1版
印数：1－4 000册　　2013年12月河北第1次印刷

定价：69.00元

读者服务热线：(010)51095186转600　印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京崇工商广字第 0021 号

站在巨人的肩上

Standing on Shoulders of Giants

www.ituring.com.cn

序一

没有用过 Node 的人，是不会相信仅凭 JavaScript 这门活跃于网页编程的脚本语言就可以驱动后端复杂的应用程序，也不会相信 Node 在开发高并发、高性能后端服务程序上也有着极大的优势。

我们在 2010 年接触 Node 的时候，国内外了解 Node 的人寥寥可数，2011 年我们已经决定在淘宝的部分生产系统中开始使用 Node。由于招募熟悉 Node 的人才是个大问题，为了树立技术品牌，我们在 2011 年年初创办 CNode 开源技术社区（CNodeJS.org），没有想到一发不可收拾。从 2011 年 4 月开始，我们走遍北京、上海、广州、深圳、杭州，甚至还到了香港，发起并且组织了多次 NodeParty 线下技术分享。为了弥补初学者没有 Node 托管环境学习测试的问题，我们还自己研发了 Node App Engine。Node 在国内深入人心，我相信与 CNode 社区有着不小的关系。

最初，Node 的爱好者大都是些喜欢探索新技术的极客。在社区，我们也认识了很多天南海北的朋友，包括朴灵。在一次上海 Node 技术分享会后，我邀请他加入了淘宝。他在淘宝工作之余继续为社区作贡献，自发为 Node 的推广做了很多事情，包括今天他呕心写了这本书，我相信这是目前质量最高的一本 Node 图书。因为中国没有几个人像朴灵一样，有机会在很多高并发的应用场景中反复实践。这绝对是一本实践性极强的技术书，不管是否学习过 Node，只有你爱好技术，都推荐你阅读它。

空无
CNode 社区创始人
阿里巴巴数据平台事业部数据交换平台总监

序二

Node诞生于2009年，天才的屌丝青年Ryan Dahl利用了Google的V8引擎打造了基于事件循环实现的异步I/O框架。也许Ryan当时选择JavaScript作为服务器开发语言，只是因为V8的性能远超其他脚本语言，但是这却成为Node成功的极其重要的因素。不仅仅是JavaScript巨大的用户群，更重要的是JavaScript之前没有任何I/O库，这使Node在开发异步I/O时不会像EventMachine、Twisted那样因与同步I/O混用而导致问题。

短短几年的时间，Node取得了巨大的成功。在开源社区GitHub上，Node高居第二。express、socket.io这样的优秀框架都有着极高的排名，NPM上的模块数量和下载量也非常惊人。更可喜的是，国内的Node社区也诞生了许多优秀的开源项目，其中node-webkit、pomelo等在国际开源社区中都产生了一定的影响力。

在企业界，Node的应用也越来越广泛。LinkedIn的移动平台已经全部从Ruby迁移到Node，机器数量缩减为原来的十分之一。像Yahoo、Microsoft这样的大公司，有好多应用已经迁移到Node了。国内的阿里巴巴、网易、腾讯、新浪、百度等公司的很多线上产品也纷纷改用Node开发，并取得了很好的效果。

朴灵是国内最早的Node开发者之一，不仅组织了CNode社区，在InfoQ发表的“深入浅出Node.js”系列文章更是对国内的Node社区产生了巨大的影响。记得我在2011年初次接触Node的时候，除了国外的几个演讲文稿，基本上没有Node相关的图书，而最让我印象深刻的，毫无疑问是朴灵的“深入浅出Node.js”系列文章。正是这一系列文章，使我们较好地理解、学习Node后，开发出了pomelo框架，也奠定了朴灵在国内Node界的地位。

如今两年过去了，国内外的Node图书也出了不少。但国内的几本书有点偏浅，即使国外的几本名气很大的书也没有让我有动力通读全书，因为内容整体上没有太大深度，对于有较久开发经验的Node开发者帮助不是很大。不过当朴灵让我审校这本书时，我觉得收获颇多。相比其他Node图书的作者，他在淘宝一线的开发经验使这本书更有深度，而他文艺青年的背景让这本书读起来极其顺畅，他的钻研精神又让这本书在理论上很有深度。例如，朴灵在微博上自称“一个能搞定回调函数嵌套的男人”还真不是吹的，在第4章中，他详细介绍了Node的各种嵌套函数过深的解决方案，例如EventProxy、Promise、async、step、wind.js等各种解决方案都有深入讲解。此外，朴灵还是EventProxy的作者，在这方面有最权威的实践经验。

朴灵是国内Node界的第一传道士，除了那一系列文章，他还在全国各地组织了NodeParty和JSConf China（2012年的沪JS和2013年的京JS），并且在微博上以各种诙谐幽默的方法宣传

Node。在各个技术大会上，我们都可以见到朴灵的身影。更强的是，朴灵在每次大会上所做的演讲很少雷同，他总是能挖掘出 Node 的方方面面，然后很认真地总结出来，以幽默的讲解让听众愉快地接受。

因此，当得知朴灵要写这本书时，我们都很兴奋。谁能比他更胜任呢？毫无疑问，这将是国内第一的 Node 图书。如今，经过一年多的等待，你们终于有机会看到朴灵这一年多辛勤劳动的成果了。

谢骋超
网易高级技术专家、架构师
pomelo 开源游戏服务器框架创始人
2013 年 7 月 8 日

前　言

2006年至今，我们时常可以看到JavaScript的新闻，刚开始只是JavaScript引擎性能的提升，到后来发现很多是来自HTML5和Node创造的奇迹。如果只看表面，很容易让人感觉这又是一颗卫星。这种现象让人觉得不可信，所以出现了以下各种版本的误解。

- Node肯定是几个前端工程师在实验室里捣鼓出来的。
- 为了后端而后端，有意思吗？
- 怎么又发明了一门新语言？
- JavaScript承担的责任太重了。
- 直觉上，JavaScript不应该运行在后端。
- 前端工程师要逆袭了。

一方面，大家看到JavaScript在各个地方放出异彩，其他语言的开发者既羡慕它的成果，又担心它对当前所从事的语言造成冲击；另一方面，人们还是有JavaScript只能做前端脚本的定势思维。究其原因，还是因为人们缺乏历史观层次上的认知，所以会产生一些莫须有的惴惴不安。

1995年，JavaScript随网景公司发布的Netscape Navigator 2.0发布，它最早命名为LiveScript，随后更名为JavaScript。它出自如今的Mozilla公司的CTO——Brendan Eich之手，其产生来源于网景公司发布的Netscape Navigator浏览器需要一种脚本语言来协助浏览器做一些简单的动态操作。当时网景公司与Sun公司合作密切，不懂技术的管理层希望得到一个Java的脚本版语言，以期能像Java一样风靡。Brendan Eich原本进入网景公司是希望做Scheme语言的开发，但是却接到了一个不喜欢的任务，但迫于当时形势，不得不完成此事，于是JavaScript之父在10天的时间里仓促完成了JavaScript的设计，当时的项目代号是Mocha，名字叫LiveScript。

这门语言除了看起来像Java外，本质与Java语言相去甚远，管理层期望的Java Script其实借鉴了C、Scheme、Self、Java的设计。尽管仓促，但是这门语言还是借鉴了其他语言的不少优点，如函数式、原型链继承等。处于Java阴影下的这门语言获得了它最终的名字：JavaScript。至今，仍然还有许多人分不清Java与JavaScript的关系，就像分不清雷锋与雷峰塔一样。

虽然JavaScript的产生与Netscape Navigator浏览器的需求有关系，但它并非只是设计出来用于浏览器前端的。早在1994年，网景公司就公布了其Netscape Enterprise Server中的一种服务器端脚本实现，它的名字叫LiveWire，是最早的服务器端JavaScript，甚至早于浏览器中的JavaScript公布。对于这门图灵完备的语言，网景早就开始尝试将它用在后端。

随后，微软在第一次浏览器大战时，于1996年发布的IE 3.0中也包含了它的脚本语言：JScript。基于商标的原因，它叫JScript，但是与JavaScript兼容。在1997年年初，微软在它的服务器IIS 3.0中也包含了JScript，这就是我们在ASP中能使用的脚本语言。鉴于微软处处与网景针锋相对，出于保护自己的目的，网景公司推进了JavaScript的标准化进程，于1996年11月将JavaScript递交给ECMA国际标准组织，在1997年7月公布了第一个版本，是为ECMA-262号标准，又称ECMAScript。

可以看到，JavaScript一早就能运行在前后端，但风云变幻，在前后端各自的待遇却不尽相同。伴随着Java、PHP、.NET等服务器端技术的风靡，与前端浏览器中的JavaScript越来越重要相比，服务器端JavaScript逐渐式微。只剩下Rhino、SpiderMonkey用于工具。

然而，这个世界是变化的。第一次浏览器大战落幕后的JavaScript的世界有些平静，但依然在萌生一些变化。Google对Ajax的应用让JavaScript变得越来越重要。Firefox的发布掀起了对IE的反攻，迎来了第二次浏览器大战，竞争令JavaScript的性能不断提升，Chrome的加入令它高潮迭出。CommonJS规范的提出，不断在完善JavaScript。ECMAScript标准的不断推进，令语言更加精炼简洁，不停地去芜存菁。

浏览器端JavaScript在Web应用中盛行，甚至让人们忘掉了JavaScript可以在服务器端运行这码事。但是，服务器端JavaScript现在回来了，因为Node诞生了。Node的诞生离不开上述的历史契机，服务器端JavaScript在漫长的历史中长期停滞留下空白，但Node重新将这个领域激活。Ryan Dahl基于对高性能Web服务器的探索，无意间促成了服务器端JavaScript领域的焕然一新。Node凭借V8的高性能和异步I/O模型将JavaScript重新推向了一个高潮。现在，Node不仅满足JavaScript同时运行在前后端，而且性能还十分高效。与传统印象中的不同，它甚至可比于当前的高效脚本语言。

奇妙的反应还在继续，前后端要跨语言开发的现状已经开始改变，因为语言堆栈的不同，开发者的分工也进行了细分：前端工程师和后端工程师。专业技能因为分工而精进，但也将技能变为专利，似乎前端工程师不能进行后端开发，后端工程师搞不定前端开发，犹如树立的墙。但Node的出现令这种分工的界限又开始模糊了。同时一些后端工程师也关注到Node，他们甚至不关心前后端语言是否一致，而是赤裸裸地表示对Node高性能的垂涎，如实时、高并发等。

大量的前后端工程师加入了Node的开发阵营，GitHub上JavaScript是最活跃的开发语言，NPM社区第三方模块恐怖的增长速度和下载量都昭示着这个过程不可逆，在这里吼一声万能的NPM，总能找到你需要的解决方案。很多不断涌现的项目和创意都因为Node和前端开发能共用一种语言而独特。换言之，Node的本意是提供一个高性能的面向网络的执行平台，但无意间促成了JavaScript社区的繁荣，并进而形成强大的生态系统。

本书目的

目前，还没有一本书将Node自身结构介绍出来，大多停留在Node介绍或者框架、库的使用层面上，本书希望从不同的视角揭示Node自己内在的特点和结构。也许你已经用过Node进行

相关的开发，在使用了 Node 带来的欣喜后，还能在阅读本书时，发出一句“哦，原来 Node 是这样的”，这就是本书的简单寄望。

对于 Node 初学者，目前市面上也已经有 Node 相关的入门书，它们可以快速地领你进入 Node 开发之旅。在了解了这些基本过程后，想了解更多 Node 知识的好奇心，会领你来阅读本书的。

阅读建议

本书并非完全按照顺序递进式介绍，如第 2 章是从代码组织结构看待 Node，第 3 章是从运行结构看 Node，第 4 章则是从编程结构看 Node，第 5 章则是 Node 中内存结构的揭示，第 6 章谈及的是 Node 中的数据在 I/O 流中的结构或状态，第 7 章是 Node 在网络服务角度的介绍，第 8 章是 Node 在 HTTP 上的展现，第 9 章讨论了 Node 的单机集群结构，第 10 章是从单元测试和性能测试的角度去关注 Node，第 11 章虽然已经脱离了 Node 编码的范畴，但是站在产品化的角度看待 Node，也会颇有收获。

下面是各章的详细介绍。

第 1 章：这一章简要介绍了 Node，从中可以了解 Node 的发展历程及其带来的影响和价值。

第 2 章：这一章介绍了 Node 的模块机制，从中可以了解到 Node 是如何实现 CommonJS 模块和包规范的。在这一章中，我们详细解释了模块在引用过程中的编译、加载规则。另外，我们还能读到更深度的关于 Node 自身源代码的组织架构。

第 3 章：这一章展示了在 Node 中我们将异步 I/O 作为主要设计理念的原因。另外，还会介绍到异步 I/O 的详细实现过程。

第 4 章：这一章主要介绍异步编程，其中有常见的异步编程问题介绍，也有详细的解决方案。在这一章中，我们可以接触到 Promise、事件、高阶函数是如何进行流程控制的。

第 5 章：这一章主要介绍了 Node 中的内存控制，主要内容有垃圾回收、内存限制、查看内存、内存泄漏、大内存应用等细节。

第 6 章：这一章介绍了前端 JavaScript 里不能遇到的 Buffer。由于 Node 中会涉及频繁的网络和磁盘 I/O，处理字节流数据会是很常见的行为，这部分场景与纯粹的前端开发完全不同。

第 7 章：这一章介绍了 Node 支持的 TCP、UDP、HTTP 编程，还附赠了 WebSocket 与 TLS、HTTPS 的介绍。

第 8 章：这一章介绍了构建 Web 应用的过程中用到的大多数技术细节，如数据处理、路由、MVC、模板、RESTful 等

第 9 章：这一章介绍了 Node 的多进程技术，以及如何借助多进程的方式来提升应用的可用性和性能。

第 10 章：这一章介绍了 Node 的单元测试和性能测试技巧。

第 11 章：“行百里者半九十”，完成产品开发的代码编写后，才完成了项目的第一步。这一

章介绍了将 Node 产品化所需要注意到的细节，如项目工程化、代码部署、日志、性能、监控报警、稳定性、异构共存等。

附录 A：详细介绍了 Node 的安装步骤。

附录 B：讨论了 Node 的调试技巧。

附录 C：探讨了团队实践或多人协作过程中需要关注的编码规范问题，它可以很好地规避一些低级的、明显的错误。

附录 D：作为企业开发者，必须关注模块仓库的搭建管理。在这一章中，我们介绍了如何通过搭建私有 NPM 来解决企业隐私安全等方面的问题。

致　谢

这本书的产出过程其实完全不在意料之中。最早找到我的杨海玲老师当初还在图灵公司，那还是 2011 年的时候，作为 Node 发烧友，我其实是极度心虚的，因为我除了作为前端工程师所拥有的那点 JavaScript 知识外，只有学习 Node 的热情，当时我十分感动，然后拒绝了杨老师的邀请。

随后，崔康老师在 CNode 社区看到我的那篇“用 Node.js 打造你的静态文件服务器”后，邀请我加入他在 InfoQ 上开辟的“深入浅出 Node.js”专栏，出于对写作的恐惧，我也拒绝了崔康老师的邀请。崔康老师随后以“写专栏只要每个月写点，远比写书容易”的理由劝服我，我随即在心中拿捏了计划，觉得可以将自己的学习经验写出来，边学边写，前前后后大概可以写出许多东西来，于是答应了崔康老师。在随后的大半年时间里，我在 InfoQ 上发表了 7 篇专栏文章。可能是圈子太小，杨老师在寻找 Node 原创书作者的过程中经过一圈又从崔康老师的推荐下回到了我这里。因为心中已经有些眉目，知道自己想要表达些什么，加上加入阿里巴巴数据平台数据产品部门（EDP）专职从事 Node 开发后，团队的领导玄澄和苏千都十分鼓励我，觉得这使命冥冥之中该由我去完成，于是应承了这本书的写作。

当然，这只是苦逼日子的开始，尽管每天接触的还是 JavaScript 语言，但实际上已经从前端领域进入了后端领域，我的知识面远远不足以支撑这本书的写作。跨领域的过程是相当痛苦的，很少有人喜欢尝试改变已有的习惯，而我还要在这个基础上将我还不太熟悉的东西重新分享出来，要保证没有错误，这是远比专栏写作高得多的挑战，为此我屡次有上了贼船的感觉。直觉上，因为 Node 是 JavaScript 语言，所以前端工程师掌握它是相对容易的，但是事实上，“行百里者半九十”，熟悉 JavaScript 只是帮助我少了十里路，在整个历程中，还有九十里需要完成，这就是兴趣与现实之间的差距。

经历了拖稿、延期以及因为没能按期出版而输掉 iPad 奖励等打击，最终梳理出了这本书的内容。与大多数介绍 Node 的书不同，这些内容的写作过程就是我自己学习 Node 的过程，这个过程充斥了改变带来的痛苦和收获，每一章讲述的侧重点都不相同，但又都是 Node。我在这个过程中完成了自己在操作系统、网络方面的知识补充，蜕变的过程总是寂寞和喜悦的，过去因为前后端语言的不同而分散疏离的知识点，奇迹般地因为 Node 重新组合连接起来，这大概就是乔布斯提到的“connecting the dots”吧。写完这本书时，我前端工程师的职位名已经被老板摘掉，姑且认为是玄澄对我转变过程的认可。

最后，非常感谢王军花老师跟进本书的进度，感谢 CNode 社区的朋友们提出宝贵建议，感谢阿里巴巴 EDP 部门给予我最好的环境去成长，让这本书更精彩。

想不到曾经以文艺青年自诩的我，以这样的形式完成了一本书的写作，既在意料之外，也在意料之中。这本书也不能用来致青春，这里献给我的母亲，没有您的影响，不可能存在这本书。

目　录