《深入理解Nginx》

图书基本信息

书名：《深入理解Nginx》

13位ISBN编号：9787111414780

10位ISBN编号：7111414780

出版时间：2013-4-15

出版社：陶辉 机械工业出版社 (2013-03出版)

作者：陶辉

页数：584

《深入理解Nginx》

前言

为什么要写这本书当我试图在产品的关键位置设计一个高性能Web服务器时，我选择使用成熟的Nginx。选择它的理由为：首先，它对服务器性能上的挖掘已经达到了很高水平，它能尽量使不同的硬件（包括网卡、硬盘、不同的CPU核心）并发运行，同时软件中又没有阻塞进程使之睡眠的代码，从性能上来说，它可以挑战任何服务器。其次，完全基于事件驱动的服务器开发效率往往很不理想，它们要处理的事件过于底层化、细节化，这使得各功能模块无法聚焦于业务，最终产品的功能都较为单一，不会有丰富的可选功能。但Nginx却不然，由于它在软件架构上具有优秀的设计，使得Nginx完全由许多简单的模块构成，各模块（特别是HTTP模块）不用介入底层细节，在尽享分阶段、无阻塞的事件驱动架构下，可以专注于业务功能的实现，这样最终为Nginx带来了大量的官方、第三方的功能模块，使得功能同样强大的Nginx在产品核心位置上足以担当重任，经受住海量请求的考验。当Nginx已有模块提供的功能不能完全实现我的所有业务需求时，我可以在Nginx的后端再搭建一个实现了缺失功能的非Nginx服务器，将Nginx无法实现的请求反向代理到这台服务器上处理。但这样也有一定的弊端，首先增大了处理请求的开销，其次后端服务器的设计仍然制约着总体性能（它依然需要解决Nginx解决过的无阻塞问题，那样才能像Nginx一样高效），这样做仅适用于对性能要求不高的场景。唯有开发一个实现了所需功能的自定义Nginx模块嵌入到Nginx代码中，才能让自己的业务像Nginx一样充分挖掘服务器的硬件资源，及时地响应百万级别的并发TCP连接。当我在开发Nginx模块之前，试图在市面上找到一本关于Nginx模块开发的书籍（无论是中文还是英文）时却一无所获。我只能找到如何使用Nginx及其已有模块的书籍。为了开发Nginx模块，我只能通过阅读Nginx极度缺少注释的源代码，并分析各种官方Nginx模块来逐步还原其设计思想，反复尝试、验证着怎样的模块能够使用Nginx的基础架构，和丰富的跨平台工具方法，同时符合Nginx设计思想，使Nginx拥有媲美Linux内核的一流效率。这个过程耗费了我很多的精力，因此，我希望今后的Nginx使用者、开发者在遇到同样的问题时，不至于还要很痛苦地阅读源代码来找到模块开发方法，而是简单地按照章节查阅本书，就可以快速找到怎样简单、高效地开发Nginx模块，把精力放在业务的实现上。这是我写这本书的第一个目的。当我们产品中运行的Nginx出现了问题时，往往是通过找到错误的配置项、使用方式来解决的，这样也的确能够修复大部分问题。但是更深层次的问题，或者是使用场景比较偏僻，抑或是Nginx自身代码考虑得不够全面时，这些问题往往只能由那些花费大量精力研究Nginx源代码的工程师来解决。我写作本书的第二个目的是希望通过透彻地解析Nginx架构，帮助读者深入理解Nginx，既能够正确地使用它，也能在它出现任何问题时找到根本原因，进而用最合适的方法修复或者回避问题。Nginx是一个优秀的事件驱动框架，虽然它在HTTP的处理上非常出色，但它绝不仅仅用于Web服务器。Nginx非常适合开发在传输层以TCP对外提供服务的服务器程序。基于Nginx框架开发程序有5个优势：1）Nginx将网络、磁盘及定时器等异步事件的驱动都做了非常好的封装，基于它开发将可以忽略这些事件处理的细节。2）Nginx封装了许多平台无关的接口、容器，适用于跨平台开发。3）优秀的模块化设计，使得开发者可以轻易地复用各种已有的模块，其中既包括基本的读取配置、记录日志等模块，也包括处理请求的诸如HTTP、mail等高级功能模块。4）Nginx是作为服务器来设计其框架的，因此，它在服务器进程的管理上相当出色，基于它开发服务器程序可以轻松地实现程序的动态升级，子进程的监控、管理，配置项的动态修改生效等。5）Nginx充分考虑到各操作系统所擅长的“绝活”，能够使用特殊的系统调用更高效地完成任务时，绝不会去使用低效的通用接口。尤其对于Linux操作系统，Nginx不遗余力地做了大量优化。当我们期望编写一款能够以低负载处理高并发请求并且主要处理基于TCP的服务器程序时，推荐选择Nginx，它可能会带给我们意外的惊喜。这本书的第三部分，将通过分析Nginx的内部架构，帮助读者了解怎样基于Nginx开发高效的TCP服务器程序：通过开发一种新的模块类型，实现一种新的功能框架来提供极佳的扩展性，使得功能子模块仅关注于业务的开发，忽视底层事件的处理。这是我写作本书的第三个目的。除了这3个主要目的外，我还希望通过这本书向大家展示Nginx在服务器开发上的许多巧妙设计，它们或在抽象设计上精妙，或通过操作系统精确、节省地使用硬件资源，这些细节之处的设计都体现了Igor Sysoev的不凡功底。即使我们完全不使用Nginx，学习这些技巧也将有助于我们服务器编程水平的提升。读者对象本书适合以下读者阅读。对Nginx及如何将它搭建成一个高性能的Web服务器感兴趣的读者。希望通过开发特定的HTTP模块实现高性能Web服务器的读者。希望了解Nginx的架构设计，学习怎样充分使用服务器上的硬件资源的读者。了解如何快速定位、修复Nginx中深层次Bug的读者。希望利用Nginx提供的框架，设计出任何基于TCP的、无阻塞的、

易于扩展的服务器的读者。背景知识如果仅希望了解怎样使用已有的Nginx功能搭建服务器，那么阅读本书不需要什么先决条件。但如果希望通过阅读本书的第二、第三部分，来学习Nginx的模块开发和架构设计技巧，则必须了解C语言的基本语法。在阅读本书第三部分时，需要读者对TCP有一个基本的了解，同时对Linux操作系统也应该有简单的了解。如何阅读本书我很希望将本书写成一本“step by step”式（循序渐进式）的书籍，因为这样最能节省读者的时间，然而，由于3个主要写作目的想解决的问题都不是那么简单，所以这本书只能做一个折中的处理。在第一部分的前两章中，将只探讨如何使用Nginx这一问题。阅读这一部分的读者不需要了解C语言，就可以学习如何部署Nginx，学习如何向其中添加各种官方、第三方的功能模块，如何通过修改配置文件来更改Nginx及各模块的功能，如何修改Linux操作系统上的参数来优化服务器性能，最终向用户提供企业级的Web服务器。这一部分介绍配置项的方式，更偏重于带领对Nginx还比较陌生的读者熟悉它，通过了解几个基本Nginx模块的配置修改方式，进而使读者可以通过查询官网、第三方网站来了解如何使用所有Nginx模块的用法。在第二部分的第3章～第7章中，都是以例子来介绍HTTP模块的开发方式的，这里有些接近于“step by step”的学习方式，我在写作这一部分时，会通过循序渐进的方式使读者能够快速上手，同时会穿插着介绍其常见用法的基本原理。在第三部分，将开始介绍Nginx的完整框架，阅读到这里时将会了解第二部分中HTTP模块为何以此种方式开发，同时将可以轻易地开发出Nginx模块。这一部分并不仅仅满足于阐述Nginx架构，而是会探讨其为何如此设计，只有这样才能抛开HTTP框架、邮件代理框架，实现一种新的业务框架、一种新的模块类型。对于Nginx的使用还不熟悉的读者应当从第1章开始学习，前两章将帮助你快速了解Nginx。使用过Nginx，但对如何开发Nginx的HTTP模块不太了解的读者可以直接从第3章开始学习，在这一章阅读完后，即可编写一个功能大致完整的HTTP模块。然而，编写企业级的模块必须阅读完第4章才能做到，这一章将会介绍编写产品线上服务器程序时必备的3个手段。第5章举例说明了两种编写复杂HTTP模块的方式，在第三部分会对这两种方式有进一步的说明。第6章介绍一种特殊的HTTP模块—HTTP过滤模块的编写方法。第7章探讨基础容器的用法，这同样是复杂模块的必备工具。如果读者对于普通HTTP模块的编写已经很熟悉，想深入地实现更为复杂的HTTP模块，或者想了解邮件代理服务器的设计与实现，或者希望编写一种新的处理其他协议的模块，或者仅仅想了解Nginx的架构设计，都可以直接从第8章开始学习，这一章会从整体上系统介绍Nginx的模块式设计。第9章的事件框架是Nginx处理TCP的基础，这一章无法跳过。阅读第8、第9章时可能会遇到许多第7章介绍过的容器，这时可以回到第7章查询其用法和意义。第10章～第12章介绍HTTP框架，通过这3章的学习会对HTTP模块的开发有深入的了解，同时可以学习HTTP框架的优秀设计。第13章简单地介绍了邮件代理服务器的设计，它近似于简化版的HTTP框架。第14章介绍了进程间同步的工具。为了不让读者陷入代码的“汪洋大海”中，在本书中大量使用了图表，这样可以使读者快速、大体地了解流程和原理。关键地方会直接给出代码，并添加注释加以说明。希望这种方式能够帮助读者减少阅读花费的时间，更快、更好地把握Nginx，同时深入到细节中。在本书开始写作时，由于Nginx的最新稳定版本是1.0.14，所以本书是基于此版本来编写的。截止到本书编写完成时，Nginx的稳定版本已经上升到了1.2.4。但这不会对本书的阅读造成困扰，因为本书主要是在介绍Nginx的基本框架代码，以及怎样使用这些框架代码开发新的Nginx模块，而不是介绍Nginx的某些功能。在这些基本框架代码中，Nginx一般不会做任何改变，否则已有的大量Nginx模块将无法工作，这种损失也是不可承受的。而且，Nginx框架为具体的功能模块提供了足够的灵活性，修改功能时很少需要修改框架代码。Nginx是跨平台的服务器，然而这本书将只针对最常见的Linux操作系统进行分析，这样做一方面是篇幅所限，另一方面则是本书的写作目的主要在于告诉读者如何基于Nginx编写代码，而不是怎样在一个具体的操作系统上修改配置来使用Nginx。因此，即使本书以Linux系统为代表讲述Nginx，也不会影响使用其他操作系统的读者阅读，因为操作系统的差别对阅读本书的影响实在是非常小。勘误和支持由于作者的水平有限，加之编写的时间也很仓促，书中难免会出现一些错误或者不准确的地方，恳请读者批评指正。为此，我特意创建了一个在线支持与应急方案的二级站点。读者可以将书中的错误发布在Bug勘误表页面中，同时如果你遇到任何问题，也可以访问Q&A页面，我将尽量在线上为读者提供最满意的解答。书中的全部源文件都将发布在这个网站上，我也会将相应的功能更新及时发布出来。如果你有更多的宝贵意见，也欢迎你发送邮件至我的邮箱，期待能够听到读者的真挚反馈。致谢我首先要感谢Igor Sysoev，他在Nginx设计上展现的功力令人折服，正是他的工作成果才让本书的诞生有了意义。lisa是机械工业出版社华章公司的优秀编辑，非常值得信任。在这半年的写作过程中，她花费了很多时间、精力来阅读我的书稿，指出了许多文字和格式上的错误，她提出

的建议大大提高了本书的可读性。在这半年时间内，一边工作一边写作给我带来了很大的压力，所以我要感谢我的父母在生活上对我无微不至的照顾，使我可以全力投入到写作中。繁忙的工作之余，写作又占用了休息时间的绝大部分，感谢我的太太对我的体谅和鼓励，让我始终以高昂的斗志投入到本书的写作中。感谢我工作中的同事们，正是在与他们一起工作的日子里，我才不断地对技术有新的感悟；正是那些充满激情的岁月，才使得我越来越热爱服务器技术的开发。谨以此书，献给我最亲爱的家人，以及众多热爱Nginx的朋友。陶辉

《深入理解Nginx》

内容概要

本书是阿里巴巴资深Nginx技术专家呕心沥血之作，是作者多年的经验结晶，也是目前市场上唯一一本通过还原Nginx设计思想，剖析Nginx架构来帮助读者快速高效开发HTTP模块的图书。
本书首先通过介绍官方Nginx的基本用法和配置规则，帮助读者了解一般Nginx模块的用法，然后重点介绍如何开发HTTP模块（含HTTP过滤模块）来得到定制的Nginx，其中包括开发一个功能复杂的模块所需要了解的各种知识，如Nginx的基础数据结构、配置项的解析、记录日志的工具以及upstream、subrequest的使用方法等。在此基础上，综合Nginx框架代码分析Nginx的架构，介绍其设计理念和技巧，进一步帮助读者自由、有效地开发出功能丰富、性能一流的Nginx模块。

《深入理解Nginx》

作者简介

陶辉，毕业于西安交通大学计算机科学与技术专业，曾就职于华为中央软件部、腾讯QQ空间、思科中国CRDC等公司，目前在阿里巴巴云计算公司的飞天团队工作，研究方向为介于IaaS和PaaS间的弹性计算，多年以来专注于Nginx的定制化应用，对Nginx的设计与特性有深刻认识，实战经验丰富，编写过许多优秀的Nginx模块并应用于企业级产品中，同时撰写了大量关于Nginx的技术文章。擅长Linux环境下高性能服务器的开发，以及分布式环境下海量数据存储的设计开发。

《深入理解Nginx》

书籍目录

章节摘录

版权页： 插图： 3）如果handler方法返回NGX_DONE，则意味着刚才的handler方法无法在这一次调度中处理完这一个阶段，它需要多次的调度才能完成。注意，此时返回NGX—OK，它会使得HTTP框架立刻把控制权交还给epoll等事件模块，不再处理当前请求，唯有这个请求上的事件再次被触发时才会继续执行。 4）如果handler方法返回除去NGX_DECLINED或者NGX_DONE以外的其他值，则调用ngx_http_finalize_request结束请求，其参数为handler方法的返回值。 可以注意到，ngx_http_core_rewrite_phase方法与ngx_http_core_generic_phase方法有一个显著的不同点：前者永远不会导致跨过同一个HTTP阶段的其他处理方法，就直接跳到下一个阶段来处理请求。原因其实很简单，可能有许多HTTP模块在NGX HTTP SERVERREWRITE_PHASE和NGX_HTTP_REWRITE_PHASE阶段同时处理重写URL这样的业务，HTTP框架认为这两个阶段的HTTP模块是完全平等的，序号靠前的HTTP模块优先级并不会更高，它不能决定序号靠后的HTTP模块是否可以再次重写URL。因此，ngx http corerewrite_phase方法绝对不会把phase_handler直接设置到下一个阶段处理方法的流程中，即不可能存在类似下面的代码。 11.6.3 ngx http core access phase ngx_http_core_access—phase方法是仅用于NGX_HTTP_ACCESS_PHASE阶段的处理方法，这一阶段用于控制用户发起的请求是否合法，如检测客户端的IP地址是否允许访问。它涉及nginx.conf配置文件中satisfy配置项的参数值，见表11—2。 对于表11—2的any配置项，是通过ngx_http_request_t结构体中的access_code成员来传递handler方法的返回值的，因此，ngx_http_core_access_phase方法会比较复杂，如图11—10所示。

《深入理解Nginx》

编辑推荐

《深入理解Nginx:模块开发与架构解析》由阿里巴巴资深Nginx专家撰写，透彻解析Nginx架构，详解Nginx模块开发方法和技巧。

《深入理解Nginx》

名人推荐

从耐心帮助读者了解“如何阅读本书”到书中详细的代码解析与注释、大量而精致的各种图表，以及为本书开发的在线支持网站等，无不体现了作者全身心投入本书的写作。更为可贵的是，作者在实际工作中一行行阅读Nginx源代码、不断尝试和探索，耗费了巨大的精力所积累的宝贵经验，无私地和读者分享。这也成就了Nginx这方面的巨作——覆盖了Nginx的安装、配置、模块开发、架构解析和深度应用等各个方面，适合不同层次的读者，并能切实地帮助读者有效地解决Nginx应用中所碰到的困惑与难题。——朱少民 同济大学软件学院教授这是国内（或许也是国外）第一本关于Nginx模块开发的书籍。作者有着丰富的Nginx开发和运维经验，其定制的Nginx服务于真实的大并发SaaS应用，因此其作品不是一本泛泛的手册，而是经验教训之心血结晶。这本书的面世，对于致力于打造符合自己应用场景之高性能Web服务器的开发人员来说，无疑是一大福音。——Grant pan, 思科CRDC Senior Manager在互联网上，关于如何安装及配置Nginx的文章可谓众多，可惜一直以来却缺少面向开发人员对其架构原理及核心模块进行系统阐述的相关著作。本书面向不同层次的读者，对Nginx的使用、配置、架构原理及模块开发进行了系统而细致的阐述，无论是单纯使用Nginx的系统工程师还是专注于高性能服务器端研发的开发人员，都可以在本书中发现你所需要的内容。向Igor Sysoev致敬!感谢陶辉为我们带来这本很棒的书！——范昕 思科（美国），Senior EngineerNginx是一个功能丰富、插件（模块）众多、性能卓越的Web服务器，业界多把它放在业务的最前端充当静态资源服务器或者反向代理服务器，应用广泛。本书循序渐进地揭开了Nginx的面纱，从如何使用原生的Nginx入手，进而以几个简明的例子为主线说明如何开发http模块，最后再综合介绍Nginx的完整设计思路，帮助读者快速、深入地掌握如何基于Nginx开发出高性能服务器。——吴峥涛 阿里巴巴云计算公司 架构师

《深入理解Nginx》

精彩短评

1、看了5周，很适合作为研究Nginx源码的参考
2、先后看了两遍，第一遍主要看自定义模块，写一个http module和filter，其他部分就看的比较吃力了。
看了一段时间的源码后再看第二遍，2/3基本理解了，还有upstream比较难。

书总体写的还不错，把主流程和难点说的比较清楚，可以节约不少时间。根本解决还是源码.
3、本来主要是想学习一下 nginx 的工作原理， 部分实现细节， 配置， 部署，运维等。但是没想到是一本二次开发的书。主要讲到，用c/c++ 开发 nginx tcp,http 等网络服务 模块，但不得不说这本书是一本好书。
4、书内容很不错，讲的很好
5、用它来辅助我读代码的，三个月了，代码总算要读完了
6、带我走进了nginx的世界
7、很赞很不错，哈哈
8、计算机方面的书我一般都会优先看国外的经典，这本是少有的例外。
9、基于源码的细致解析，适用于模块开发人员，有近一半内容没怎么看懂，我滚去啃 openresty 了……
10、这本书深入浅出,喜欢
11、绝对是大作，只不过后面的部分自己没有涉及到或还没到那个程度。留着需要多看几篇。
12、还凑合，手里头废话比较多，只是国内国外这类的书比较少。看了一遍，实质性的东西不多，压缩下没多少东西，大部分都是车轱辘话。
13、解释得很全面看了张宴的再看这个非常OK
14、基本上这本书涵盖的点都是我想关注了解的，也都是nginx最核心的知识点。缺陷就是代码注释太多，讲解思路有点天马行空。看源码分析的书，还是不能太依赖作者，得自己一边看着相关的代码，再看别人怎么说的，带着问题看以我为主，能收获的更多。
15、干总让看的书，边看边开发，nginx挺强大的
16、非常好的书籍，准备深入学习
17、只读了前两章，够简单使用了。后面的太深，需要的时候再看吧。
18、大致翻了一遍,以后有机会还要深入读
19、还不错，只是最近浮躁，太技术的部分不想看
20、太高深，暂时读不懂
21、awesome. 开荒级别书籍，不过有些简略的地方太简略了还不如不说
22、部分章节很给力。
23、好书！nginx方面绝对的大师之作！可是它不太适合我，因为我只是想了解nginx基本的功能以及特性，满满的自定义模块让我有恐慌感。如果只是入门的，联系换别的书吧。
24、这本书写的很有条理，看出作者的用心，读得爽！很多离散的知识点被这本树梳理起来了
25、赶脚少点啥呢
26、这应该是讲nginx模块开发最为经典的书籍啦
27、真的是_架构_解析。。只有架构层次的解析，细节还得自己看，流程图很有帮助，就是kindle的epub格式用来贴代码简直是屎啊。。。
28、例子太少了. 参考书而已
29、书的内容很详尽，是用心写的书
30、写的还不错，深度也够。
31、还没开始看，但还是很不错
32、不错的一本Nginx书，实践+原理分析，
33、对理解Nginx工作原理很有帮助
34、不过市面上 互联网上相关内容都太少了 可以理解 还不错
35、主要是nginx的c模块继承，由于是直接编译进入nginx，具有较高的性能。同时有较大的耦合和难

度
36、很好的一本Nginx书籍，很早之前闻名Nginx的nb，去年看了这本书才接触的Nginx，确实很nb，最近又有Nginx上开发的项目，会经常翻这本书
37、让我更深入的学习Nginx
38、正在看，买的是电子版，放在kindle上看，配合着Mastering Nginx这本pdf看的，后续会继续补充读后感，哈哈奇怪的是，卓越为啥不把纸质+电子捆绑着卖呢？不至于没想到吧。。。
39、看了大部分章节，还需要实战
40、这本书极其符合我的胃口所以给满5好评，因为我就是想要Nginx的所有流程细节，这本书十分细致、『啰嗦』地一一讲解。不一定所有人都能慢慢啃完，因为代码细节、流程图非常多，想迅速看完很快就会被枯燥打击得毫无信心。建议每天抽一些固定时间（例如我是在地铁上）每天看一点，到看完的时候，你就对Nginx看法有大致把握，并且真正投入开发的时候可以以此书作为工具书进行参考。
41、配合源码看书，多思考，还有淘宝技术团队的技术博客，三管齐下，受益匪浅
42、市面上为嘛这么少nginx模块开发的书？
43、粗略看完了，比较详细的部分 待看源码时不懂时再回顾
44、其实300页就够了，其他源码实现大部分都可以省略
45、只看了配置
46、编程
47、2015.09.16, 酒仙桥, 北京
48、前段时间基本搞定waf，于是就有时间看书了。这本书非常适合用于学习Nginx源码，受益匪浅。不过upstream和mail模块还没看，暂时用不到，以后再说了。
49、内容全面，就是可读性有点差，大部分都像是代码注释
50、这部书确实不错 讲的比较深入比较适合做类似协议栈架构的开发者
51、有点难，打算重读一遍。
52、开始学习架构了
53、很不错的一本书，对nginx的全局架构把握的很到位，支持一下
54、不太喜欢写作的风格，随意翻了翻，基本大篇幅的贴源码，加中文注释。
55、nginx进阶的最好书籍一直，我略懂了一点，内容深度在我目前的理解能力之上，先看基本，初读的一遍，看完基本入门级别的，再回来重新看
56、感觉作者还是很厉害的，但是我自己的阅读体验不是特别的好。原因，1：我自己目前是个使用者，而非开发者；2：作者对书籍内容的编排不是特别好，阅读中有点累（即使是从使用者角度）。
57、虽然读过，但是在开发方面没有实践。对此书的感觉是讲解非常细致，作者的水平也毋庸置疑，其他方面没有资格评价。
58、:无

《深入理解Nginx》

精彩书评

1、这本书极其符合我的胃口所以给满5好评，因为我就是想要Nginx的所有流程细节，这本书十分细致、『啰嗦』地一一讲解。不一定所有人都能慢慢啃完，因为代码细节、流程图非常多，想迅速看完很快就会被枯燥打击得毫无信心。建议每天抽一些固定时间（例如我是在地铁上）每天看一点，到看完的时候，你就对Nginx看法有大致把握，并且真正投入开发的时候可以以此书作为工具书进行参考。
2、在苦于看源码时，想寻求一本讲解架构模块设计的书，真是困了就有枕头啊。这本书含概了nginx整体架构的设计理论及先进技巧，看了之后顿时拍手叫好。不过，可能是出版仓储，文中的印刷错误还是比较多的，需要结合源码仔细比较。这几天看了前两部分，对nginx的真心膜拜，有太多的优秀设计，收获巨多。贴出自己发现的错误：[1] P77 源码打印输出部分"%*s",应该改为“%.*s”，不然不会按照str[i]的长度输出。如果改用ngx_vslprintf可以继续使用"%*s",参考P153 表4-8。ngx自己的fmt不支持"%.*s"[2] P97 所有的“extern”都应该改为"exten" 在P96 ngx_http_request_s声明中以及源码中均声明为“exten”。[3]P97 倒数第5行，对"http_protocol指向用户请求中的HTTP的起始地址"表述感到怀疑，因为在P96 ngx_http_request_s声明中 http_protocol为ngx_str_t类型。个人认为改为“http_protocol的data域指向用户请求中的HTTP的起始地址，len存储协议版本的字符串长度”更好些。也可能是我误解了，请拍砖 :-P[4]P101 倒数第2行 "rc >= NGX_http_SPECIAL_RESPONSE"应该为 "rc >= NGX_HTTP_SPECIAL_RESPONSE", 该宏在ngx_http_request.h中77行定义：#define NGX_HTTP_SPECIAL_RESPONSE 300[5]P102 倒数第7行“而 ngx_http_request_t 的返回值是多样的”，根据上下文理解，个人感觉应该改为“而 ngx_http_send_header 的返回值是多样的”.[6]P103 第2行“如同headers_in,ngx_http_request_t也有一个headers_out"。这个地方个人无法理解，求大神解释；headers_in 是ngx_http_request_t的一个成员，而headers_in 的类型是ngx_http_headers_in_t ,该类型中没有headers_out成员。。。[7]P122 表4-2 ngx_conf_set_size_slot 对应的行为中 "K,就表示Kiobyt"应该改为“"K,就表示Kiobyte”。[8]P176 下方源码部分的注释部分中所有的“表4-7”应该改为“表4-8”[9]P321源码部分处理写事件部分的注释“//判断这个读事件是否为过期事件”感觉应该改为“//判断这个写事件是否为过期事件”[10]P333 最后一行：”Nginx启动时，ngx_accept_disabled的值就是一个负数，其值为连接总数的7/8“感觉应该改为”Nginx启动时，ngx_accept_disabled的值就是一个负数，其绝对值为连接总数的7/8“[11] P119中源码的mycf的所有成员前缀test都应该为my，原因见P117的定义。以上的都不是大问题，应该是出版商印刷的问题，不影响对陶大大的崇拜。通过这本书学到了太多，也少走了不少弯路。力荐~~~注：这本书+淘宝的《Nginx开发从入门到精通》http://tengine.taobao.org/book/index.html +Emiller的《Nginx模块开发指南》http://code.google.com/p/emillers-guide-to-nginx-module-chn/wiki/NginxModuleDevGuide_CHN
3、新到我的书当当还没完成配货

4、讲得很细，才知道Nginx提供了这么多功能，第3、4、5章讲了许多Nginx模块开发时提供的实用功能，后面的章节似乎更偏向于理论设计，有些深，还在读。如果不懂C语言，不适合读此书，全篇除了前两章到处都有C语言代码。如果开发Nginx模块，这是一本必读书！

《深入理解Nginx》

章节试读

1、《深入理解Nginx》的笔记-透析nginx架构模型，很条理

在苦于看源码时，想寻求一本讲解架构模块设计的书，真是困了就有枕头啊。这本书含概了nginx整体架构的设计理论及先进技巧，看了之后顿时拍手叫好。不过，可能是出版仓储，文中的印刷错误还是比较多的，需要结合源码仔细比较。

这几天看了前两部分，对nginx的真心膜拜，有太多的优秀设计，收获巨多。

贴出自己发现的错误：

[1] P77 源码打印输出部分"%*s",应该改为“ %.*s ”,不然不会按照str[i]的长度输出。如果改用ngx_vslprintf可以继续使用"%*s",参考P153 表4-8。ngx自己的fmt不支持"%.*s"

[2] P97 所有的“ extern ”都应该改为"exten" 在P96 ngx_http_request_s声明中以及源码中均声明为“ exten ”。

[3]P97 倒数第5行，对"http_protocol指向用户请求中的HTTP的起始地址"表述感到怀疑，因为在P96 ngx_http_request_s声明中 http_protocol为ngx_str_t类型。个人认为改为“ http_protocol的data域指向用户请求中的HTTP的起始地址，len存储协议版本的字符串长度 ”更好些。也可能是我误解了，请拍砖 :-P

[4]P101 倒数第2行 "rc >= NGX_http_SPECIAL_RESPONSE"应该为 "rc >= NGX_HTTP_SPECIAL_RESPONSE",该宏在ngx_http_request.h中77行定义：#define NGX_HTTP_SPECIAL_RESPONSE 300

[5]P102 倒数第7行“ 而 ngx_http_request_t 的返回值是多样的 ”，根据上下文理解，个人感觉应该改为“ 而 ngx_http_send_header 的返回值是多样的 ”.

[6]P103 第2行“ 如同headers_in,ngx_http_request_t也有一个headers_out"。这个地方个人无法理解，求大神解释；headers_in 是ngx_http_request_t的一个成员，而headers_in 的类型是ngx_http_headers_in_t , 该类型中没有headers_out成员。。。

[7]P122 表4-2 ngx_conf_set_size_slot 对应的行为中 "K,就表示Kiobyt"应该改为“ "K,就表示Kiobyte ”。

[8]P176 下方源码部分的注释部分中所有的“ 表4-7 ”应该改为“ 表4-8 ”

[9]P321源码部分处理写事件部分的注释“ //判断这个读事件是否为过期事件 ”感觉应该改为“ //判断这个写事件是否为过期事件 ”

[10]P333 最后一行：”Nginx启动时，ngx_accept_disabled的值就是一个负数，其值为连接总数的7/8“ 感觉应该改为 ”Nginx启动时，ngx_accept_disabled的值就是一个负数，其绝对值为连接总数的7/8 “

[11] P119中源码的mycf的所有成员前缀test都应该为my，原因见P117的定义。

以上的都不是大问题，应该是出版商印刷的问题，不影响对陶大大的崇拜。通过这本书学到了太多，也少走了不少弯路。力荐~~~

注：这本书+淘宝的《Nginx开发从入门到精通》http://tengine.taobao.org/book/index.html +Emiller的《Nginx模块开发指南》http://code.google.com/p/emillers-guide-to-nginx-module-chn/wiki/NginxModuleDevGuide_CHN

2、《深入理解Nginx》的笔记-第5章 nginx upstream启动执行流程

1)nginx 主循环定期调用事件模块检查是否有网络请求

2）事件模块接收到http请求，根据配置文件交由http框架处理

3）http框架在处理请求到NGX_HTTP_CONTENT_PHASE阶段时，如果请求主机域名、URI、与mytest的配置项相匹配，调用模块 ngx_command_t ngx_http_mytest_commands中的处理函数ngx_http_mytest中

设置的handler函数“ ngx_http_mytest_handler ”
4）在处理函数中，调用ngx_http_get_module获取上下文，如果为空，则调用ngx_palloc分配内存空间，新建一个上下文并调用ngx_http_set_ctx与请求关联.
5)调用ngx_http_upstream_create方法为每一个请求创建upstream。
6）获取配置信息
7）用配置信息的结构体赋值r->upstream->conf成员（upstream的限制性参数）
8）初始化resolved结构体(可以指定上游服务器的地址)
9）把第三方服务器的地址设置到resolved结构体中。
10）设置upstream的回调函数：create_request、process_header、finalize_request
11）增加请求的引用计数：r->main->count++;
12)调用ngx_http_upstream_init方法启动upstream
13）return NGX_DONE 交还控制权给nginx

个人愚见，欢迎拍砖:-P

《深入理解Nginx》

版权说明