第一图书网, tushu007.com

<<Apache源代码全景分析第1卷>>

图书基本信息

书名：<<Apache源代码全景分析第1卷>>

13位ISBN编号：9787121084744

10位ISBN编号：7121084740

出版时间：2009年

出版时间：电子工业出版社

作者：张中庆,梁雪平

页数：604

字数：850000

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：http://www.tushu007.com

前言

如果说没有Apache就没有Intemet可能有些夸张，但至少可以说没有Apache，互联网就不会发展得这么快。
根据互联网研究公司NetCraft的统计，多年来Apache一直稳居Web服务器市场的头把交椅，至今仍占据超过50%的市场份额。
就整个互联网来说，Apache仍然是最重要的软件之一。
尽管近几年来涌现出不少以“高性能”为卖点的新的Web服务器软件，比如Lighttpd、Nginx等，吸引了不少用户注意力，不过Apache因其功能广泛，某些方面仍具有不可替代性，仍然是Web服务器技术领域的风向标。
话说回来，“重剑无锋，大巧不工”，有的时候软件性能表现不佳，更多原因可能是用户对其了解不够、使用不当造成的，并非软件自身有多大缺陷。
对Apache来说，更是如此。
因此，通过分析源代码了解Apache软件架构体系，熟知其本质，方能更有效地使用Apache Web服务器，使之发挥出最大效能。
为网站节省资源，为企业节省资金，也能为用户提供更好的访问体验，好处多多。
此外，随着互联网业务的复杂化，很多网站在使用Apache的过程中也遇到了新的挑战，如常常要在业务的驱动下对Apache进行扩展性的开发（例如扩展日志模块以便于更复杂的日志统计）。
这时，源代码分析是绕不过去的一件事儿，尽管源代码获取是轻而易举之事，但Apache代码毕竟凝聚了开源软件界的群体智慧，要想高效分析也并非易事，相信这本书能让有此需求的读者少走弯路，剥丝抽茧，获得更多启发与借鉴。
说起源代码分析，其实几年前市面上出现过一些有关此类话题的图书，但基本上是在大段源代码加上几句注释了事，读者可能会有吃到注水猪肉的感觉。
而读者对本书这一点大可放心，书中代码只是点到即止，相对环保多了。

第一图书网, tushu007.com

内容概要

本书是“ Apache源代码全景分析 ”的第1卷。
书中详细介绍了Apache的基础体系结构和核心模块的实现机制，包括配置文件、模块化结构、多任务并发，以及网络连接和请求读取，其中多任务并发体系结构是本书分析的重点，讨论了Prefork、Worker及WinNT三种MPM。
本书还着重介绍了Apache 2.0新引入的过滤器，包括过滤器的使用、实现，以及其中的数据组织形式——存储段和存储段组，剖析了Apache中常用的过滤器。
　　本书的目的是深入挖掘Apache运行背后的实现机制和模块开发的细节，适合Apache模块开发者、希望了解内部细节的Apache管理员、Web服务器开发者、大规模服务器开发者学习和阅读。

作者简介

张中庆，计算机软件与理论硕士。
拥有多年服务器端软件开发经验，关注大规模服务器设计技术，致力于开源技术的使用、分享和推广，《UNIX/Linux下curses库开发指南》第一作者。

书籍目录

第1章 Web服务器概述 1.1 WWW概述 1.1.1 Internet概述 1.1.2 超文本的概念 1.1.3 WWW的历史 1.2 HTTP服务器 1.2.1 HTTP服务器简介 1.2.2 HTTP服务器功能 1.2.3 WWW文档 1.2.4 工作方式 1.3 Apache功能 1.3.1 虚拟主机 1.3.2 内容协商 1.3.3 持续连接 1.3.4 缓存 1.3.5 访问控制和安全 1.3.6 动态内容生成第2章 Apache体系结构 2.1 Apache目录 2.2 Apache层次结构 2.2.1 操作系统支持层 2.2.2 可移植运行库层 2.2.3 核心功能层 2.2.4 可选功能层 2.2.5 第三方支持库 2.2.6 Apache工具包 2.3 Apache核心功能层 2.3.1 核心与可选模块的关系 2.3.2 核心组件 2.4 Apache运行流程 2.4.1 Apache启动过程 2.4.2 HTTP连接处理 2.4.3 请求报文读取 2.4.4 请求处理 2.4.5 内容生成 2.4.6 关闭与重启 2.5 主程序main 2.5.1 主程序概要 2.5.2 主程序细节第3章 配置文件管理 3.1 Apache配置系统 3.2 配置文件 3.2.1 配置文件类 3.2.2 配置文件处理时机 3.3 指令相关概念 3.3.1 指令概述 3.3.2 指令参数 3.3.3 指令上下文 3.3.4 指令参数类型 3.4 指令配置 3.4.1 指令结构 3.4.2 指令定义 3.4.3 预定义指令函数 3.4.4 指令表 3.5 Apache配置处理 3.5.1 指令保存 3.5.2 指令读取 3.5.3 配置指令处理 3.5.4 特殊指令 3.6 .htaccess处理 3.6.1 .htaccess使用场合 3.6.2 指令的覆盖 3.6.3 处理.htaccess 3.7 实现自己的配置段第4章 Apache模块化体系结构 4.1 Apache模块概述 4.1.1 Apache模块组成 4.1.2 Apache核心与模块交互 4.2 Apache模块结构 4.3 模块的加载 4.3.1 模块变量 4.3.2 DSO（Dynamic Shared Object，动态共享对象）的概念 4.3.3 静态模块加载 4.3.4 动态模块加载 4.3.5 模块卸载 4.4 指令表 4.4.1 指令表概述 4.4.2 指令处理函数 4.4.3 指令共享 4.5 挂钩（HOOK） 4.5.1 为什么引入挂钩 4.5.2 声明挂钩 4.5.3 挂钩数组声明（APR—HOOK—LINK） 4.5.4 挂钩结构（APR—HOOK—STRUCT） 4.5.5 挂钩函数注册（APR—IMPLEMENT—EXTERNAL—HOOK—BASE） 4.5.6 使用挂钩 4.5.7 挂钩排序 4.5.8 可选挂钩 4.5.9 挂钩纵览 4.5.10 自己编写挂钩 4.6 模块与配置文件 4.6.1 概述 4.6.2 如何描述配置信息 4.6.3 目录相关配置（Per—Directory Config） 4.6.4 服务器配置（Per-Server Config） 4.7 配置存储和使用 4.7.1 配置向量 4.7.2 配置存储体系结构 4.7.3 虚拟主机配置存储 4.7.4 目录配置存储 4.7.5 Location配置存储 4.7.6 文件配置存储 4.7.7 总体存储示意 4.8 模块通信 4.8.1 简单通信方式 4.8.2 可选函数 4.8.3 提供者API 4.9 常用模块 4.9.1 缓存模块 4.9.2 URL映射模块 4.9.3 内容生成模块 4.9.4 安全模块 4.9.5 代理模块 4.9.6 其余模块第5章 多任务并发处理 5.1 多进程并发处理 5.1.1 概述 5.1.2 MPM在Apache中的位置 5.2 MPM数据结构 5.2.1 记分板 5.2.2 终止管道（Pipe of Death） 5.3 Inetd：通用的多任务处理结构 5.3.1 服务器程序概述 5.3.2 INETD 5.4 预创建（Prefork）MPM分析 5.4.1 Leader/Follow模式 5.4.2 Prefork MPM概述 5.4.3 Prefork MPM实现 5.5 工作者（Worker）：MPM分析 5.5.1 Worker MPM概述 5.5.2 Worker主进程 5.5.3 子进程管理 5.5.4 线程管理 5.5.5 信号处理 5.6 WinNT MPM分析 5.6.1 WinNT MPM概述 5.6.2 完成端口相关概念 5.6.3 WinNT MPM主程序 5.6.4 监控主进程 5.6.5 工作进程 5.6.6 线程处理第6章 网络连接 6.1 网络连接概述 6.1.1 网络连接上下文环境 6.1.2 等待连接 6.1.3 接受连接 6.1.4 创建连接 6.2 连接数据结构 6.3 等待连接 6.3.1 概述 6.3.2 套接字创建 6.3.3 套接字侦听 6.4 连接处理 6.4.1 连接处理概述 6.4.2 创建连接 6.4.3 连接处理 6.5 请求读取 6.5.1 请求读取概述 6.5.2 HTTP请求报文 6.5.3 request_rec结构 6.5.4 请求读取实现 6.5.5 请求行读取 6.5.6 请求头读取 6.5.7 网络IO读写第7章 过滤器 7.1 过滤器概述 7.2 过滤器类型 7.3 过滤器结构 7.4 过滤器协议 7.5 过滤器使用 7.5.1 静态过滤器使用 7.5.2 动态过滤器使用 7.6 过滤器操作 7.6.1 过滤器注册概述 7.6.2 数据结构描述 7.6.3 过滤器结点 7.6.4 过滤器注册 7.6.5 过滤器的查找 7.6.6 添加过滤器至指定请求或连接 7.6.7 从连接中删除过滤器 7.6.8 过滤器初始化 7.7 智能过滤器 7.7.1 何谓智能过滤器 7.7.2 智能过滤器的使用 7.7.3 智能过滤器的实现 7.8 过滤器函数 7.8.1 输出过滤器 7.8.2 输入过滤器第8章 存储段和存储段组 8.1 什么是存储段和存储段组 8.1.1 存储段和存储段组 8.1.2 为什么需要存储段组 8.2 存储段分配子 8.2.1 概述 8.2.2 分配子创建 8.2.3 存储段内存分配 8.2.4 存储段内存释放 8.3 存储段操作概述 8.3.1 存储段接口 8.3.2 存储段空接口 8.4 存储段类型 8.4.1 堆存储段（Heap Bucket） 8.4.2 内存池存储段（Pool Bucket） 8.4.3 文件存储段

（File Bucket） 8.4.4 MMAP存储段（MMAP Bucket） 8.4.5 套接字存储段（Socket Bucket） 8.4.6 管道存储段（Pipe Bucket） 8.4.7 持久存储段（Immortal Bucket） 8.4.8 临时存储段（Transient Bucket） 8.4.9 刷新存储段（Flush Bucket） 8.4.10 流终止（EOS）存储段 8.4.11 HTTP错误存储段 8.5 存储段操作 8.6 存储段组操作 8.6.1 创建存储段组 8.6.2 存储段组的销毁 8.6.3 存储段组的分裂 8.6.4 统计存储段长度 8.6.5 存储段转换 8.6.6 数据写入 8.6.7 ap_r函数写入 8.7 存储段组和过滤器 8.7.1 存储段组和过滤器的关系 8.7.2 获取存储段组 8.7.3 存储段组传递第9章 常用过滤器 9.1 概述 9.2 输入过滤器 9.2.1 CORE_IN输入过滤器 9.2.2 HTTP_IN过滤器 9.2.3 创建自己的输入过滤器 9.3 输出过滤器 9.3.1 资源过滤器 9.3.2 内容过滤器 9.3.3 协议过滤器 9.3.4 编码转换过滤器 9.3.5 网络过滤器（CORE） 9.3.6 编写输出过滤器索引

章节摘录

第1章 Web服务器概述 1.3 Apache功能 1.3.1 虚拟主机 虚拟主机（Virtual Host）是指在一个机器上运行多个Web站点的机制 （比如：www.companyl.com和www.company2.com）。
虚拟主机的实现包括以下三种方式。
（1）Web服务器中配备多个IP地址，并且每一个逻辑Web服务器使用一个IP地址。
这种虚拟主机的实现技术被称为“基于IP”，这是最简单的虚拟主机的实现机制，但是这种机制存在一些问题，比如扩展性的问题。
一台机器所能存在的物理IP地址总是有限的，因此对于一个专门的ISP而言，如果要提供大量的虚拟主机，则会存在相当大的困难。
另外一个存在的问题就是IP地址的有限性，目前Web站点的数目远远超过IP地址的数目，因此，以IP地址区分虚拟主机，则会使Web站点的发展受到限制。
（2）Web服务器只有一个IP地址，不同的Web服务器使用不同的端口进行侦听。
因此这种服务器的请求uRI中必须明确地给出端口，而不能使用默认的Web端口80，比如http：//127.0.0.1：8900。
这种虚拟主机的实现技术可称为“基于端口”。
这种策略存在的问题是用户必须显式给出请求的端口，这对大部分用户来说显然是不太方便的。
如果忘记输入端口号或输入一个错误的端口号，则会使用错误的虚拟主机。
（3）Web服务器只有一个IP地址，同时多个域名被映射到该IP地址上。
所有的Web服务器侦听同一个端口。
服务器通过HTTP请求头中的HOST域对请求进行区分。
对于HTTP 1.1协议而言，该域是必须具备的，而低于HTTP 1.1的协议则未必如此。
因此从这个意义上说，只有HTTP 1.1协议才可以支持这种基于“HOST域”的协议。
Apache中支持上面三个方式的虚拟主机，而且通过mod_vhost_alias模块，可以使得类似的虚拟主机配置起来非常容易，减轻了管理员的负担。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:http://www.tushu007.com