图灵程序设计丛书 TURING

大规模、高性能、不间断网络服务的搭建和管理

24小时365天不间断服务

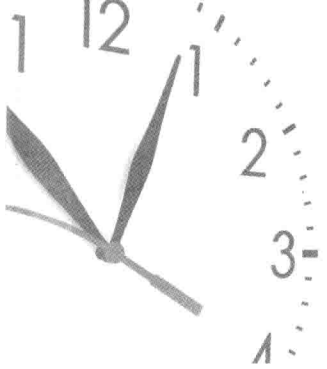
服务器/基础设施核心技术

PXE • Linux • LVS/IPVS • Puppet
Nagios • Ganglia • daemontools
he • memcached • Squid
QL • DRBD • VLAN

伊藤直也 胜见祐己
日中慎司 广濑正明 著
安井真伸 横川和哉
张毅 译

人民邮电出版社
POSTS & TELECOM PRESS

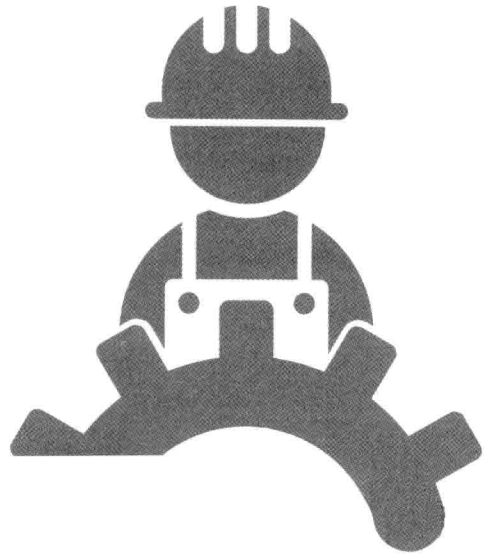
图灵i



24小时365天不间断服务

服务器/基础设施核心技术

伊藤直也 胜见祐己
[日] 田中慎司 广濑正明 著
安井真伸 横川和哉
张毅 译



人民邮电出版社
北 京

图书在版编目(CIP)数据

24小时365天不间断服务：服务器/基础设施核心技术/(日)伊藤直也等著；张毅译. -- 北京：人民邮电出版社，2015.1
(图灵程序设计丛书)
ISBN 978-7-115-38024-1

Ⅰ. ①2… Ⅱ. ①伊… ②张… Ⅲ. ①网络服务器 Ⅳ. ①TP368.5

中国版本图书馆CIP数据核字(2014)第293593号

内 容 提 要

本书是著名的网络服务供应商Hatena和Klab有限公司的工程师团队的经验总结。全书从实际的生产环境出发，就大规模、高性能、无间断的网络服务的构筑和管理技术进行了分析和说明。前3章讲解了如何搭建兼具冗余性和可扩展性的服务器/基础设施；第4章讲解了性能优化方面的内容，特别是对单个服务器的性能提升方法进行了介绍；第5章讲解了监控、管理等运行方面的内容，以笔者身边的实际生产环境为例，介绍了提升设备运行效率的技巧；第6章介绍了Hatena与KLab实际运作的网络和服务器基础设施的情况。

本书适合所有致力于运维和网络后端的开发者阅读。

◆ 著　[日]伊藤直也　胜见祐己　田中慎司
广濑正明　安井真伸　横川和哉
译　张　毅
责任编辑　乐　馨
执行编辑　杜晓静
责任印制　杨林杰
◆ 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
北京天宇星印刷厂印刷
◆ 开本：880×1230　1/32
印张：10.5
字数：292千字　2015年1月第1版
印数：1-3 500册　2015年1月北京第1次印刷
著作权合同登记号　图字：01-2012-4255号

定价：49.00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

译者序

很多人都认为网络运维是个苦差事。的确，干这行不仅要有广而全的专业知识沉淀，还需要在面对各种突发情况时做到有条不紊地沉着应对。如此，经验就成为了相关从业人员的制胜法宝。

当我第一次翻开这本书时，我异常兴奋。这本书可以说是实实在在的经验谈，虽说这些经验并非十分“前沿”和“先进”，但这些技术在实际生产环境非常接地气。此类经验谈未曾在国内成书出版，只是只言片语地零散分布在网上。可以说，这本书的出版不仅让读者看到了作者多年经验积累的结晶，而且这些经精心编排、实实在在的“干货”更有助读者付诸实践。

从编排上看，我不得不佩服原书作者和编辑的巧妙心思。这本书不但能够围绕实际运维的需要，还能从设备 / 环境搭建、性能优化、高效管理以及对未来的展望，成体系地归纳出重点知识。书中丰富的各类技巧及案例，不仅对于初学者及有经验的网络架构师难能可贵，而且也会给网络架构师带来莫大的启发。

本书还就架构设计方面，针对常见流程提出了独到的见解。例如本书坚持使用开源软件，以方便管理为主旨来定义工具，以完善的故障保护机制来应对各种灾害，重视性能与成本之间的平衡等。因此，本书不仅单纯满足于时刻不让服务中断，而且也深刻阐释了后端程序员这个角色的价值观。

在本书的翻译中，我得到了很多朋友，特别是图灵编辑的帮助和支持，在此表示深深致谢。但限于我自身能力有限，书中难免有错误及疏漏，如果读者发现了什么问题，或者有什么见解、技术想要交流，欢迎访问图灵社区搜索本书的名称，提交勘误，或者发 E-mail（Philip.Z@foxmail.com）与我联系。

张毅

2014 年于西安

关于本书

当今社会，社交软件、博客、购物网站等丰富多彩的网络服务充斥着我们的生活，E-Mail 和聊天工具更是大家常用的交流手段。可以说，互联网已经成为了我们生活中不可缺少的一部分。笔者也不例外，每天都在使用网络。更确切地说，无论公事还是私事，几乎每天都沉浸在网络中。

然而，从笔者的角度来看，与享受网络服务的终端“用户”相对应的就是服务的“提供者”。没错，笔者的工作就是网络、服务器的搭建和运营管理。

10 年前，说到网络和服务器，第一反应就是这是价格昂贵的设备，应该不是一个能够简单进入的领域。但是近些年来，随着在 PC 机上运行的 Linux、FreeBSD 等类 UNIX 操作系统的普及，以及硬件价格的降低、网络的普及，也让大家纷纷在家中建起了服务器。

这样的情况有助于我们随时获取基础设施的相关信息。特别是在部署方式和 Apache 守护程序的配置等操作方法方面，近些年的进步可谓惊人。对基础设施的新手工程师来说，这真是个方便的时代。

然而另一方面，在高效运营管理的实现、服务的冗余及可扩展等技术上的信息和技巧还远远不够。

就笔者而言，从搭建和运营数台、数十台乃至数百台的服务器系统来说，其中最大的困难就是缺少冗余和可扩展方面的信息。当时笔者还没有冗余和可扩展的相关知识和经验，完全不知该如何下手。而且想到为了实现这些还不得不使用昂贵的商用产品，连一些小小的实验也没能尝试。

现在回想起来，当时的想法是不对的。事实上，运用开源软件和常用的设备，即可搭建兼有冗余性和可扩展性的系统。但我们回过头来看看，当时迟迟无法下手的原因究竟是什么呢？难道不是单纯地因为“不知道有这个东西”“不知道可以这样”吗？

而这就是本书的写作动机。也就是说，本书的写作目标就是为读者搭建兼具冗余和可扩展性的基础设施提供启示。

本书的内容是使用开源软件的 Hatena 公司和 Klab 公司的工程师团队的经验总结，与实际运作的系统密切相关。这些信息都具有实践意义，而非夸夸其谈。系统是一个体系，是由各个要素相关联构成的。本书中不仅对每个要素的技术都进行了详细的说明，还重点介绍了各个技术要素之间的关联。但是本书并不是一本技术手册，所以并没有逐步对安装顺序进行说明，而且也不是说按照书中的命令去运行就一定能得到什么。

本书记述的是笔者在实际的开发现场所进行的思考、所面临的问题，以及为解决问题所做的努力和成果。希望在读者接下来设计、搭建和运营基础设施时，本书的内容能够为你提供参考。

作者代表　广濑正明

本书概述

本书由 6 章组成。

第 1 章　服务器及基础设施搭建入门……冗余及负载分流的基础

第 2 章　优化服务器及基础设施的拓扑结构……冗余、负载分流、高性能的实现

第 3 章　进一步完善不间断的基础设施……DNS 服务器、存储服务器、网络

1～3 章的主题是如何设计兼具冗余性以及可扩展性的基础设施。

每一章都是相互独立的，但是在“从较小的系统出发来搭建基础设施”这一大流程中，它们又是相互关联的。建议首先通读 1～3 章以把握整个流程，然后再回过头来细读感兴趣的章节。

第 4 章　性能优化、调整……Linux 单个主机、Apache、MySQL

第 4 章的主题是提升性能。

通过服务器的负载均衡来提升整个系统的性能。在这一过程中，单个服务器的性能优化是不可或缺的。第 4 章将针对单个服务器的性能可能遭遇的瓶颈，对其界定及优化方法进行介绍。

第 5 章　高效运行……确保服务的稳定提供

第 5 章的主题是监控和管理。

随着服务器数量的增加，运营成本也不断加大，那么运营成本将会成为瓶颈，进而就可能导致不能像期望中那样扩展基础设施了。换句话说，通过各种办法在最大程度上提高运行效率，是搭建具备可扩展性的基础设施的关键。第 5 章将以笔者身边的生产环境为例，来说明如何提升设备的运行效率。

第 6 章　服务后台……自律的基础设施、稳健的系统

第 6 章将对 Hatena 公司与 KLab 公司的 DSAS 实际运作的网络和服务器基础设施进行介绍。

笔者作为基础设施团队的领头人物，除了一些技术性的内容之外，还加入了前面章节中未能介绍的细节、至今为止的发展历程，以及自己作为基础设施工程师的动机和心理等非常有趣的内容，可读性很强。

章节作者一览表及出处

章节	作者
1.1 冗余的基础	安井 真伸（KLab）
1.2 实现 Web 服务器的冗余……DNS 轮询	安井 真伸
1.3 实现 Web 服务器的冗余……通过 IPVS 进行负载均衡	安井 真伸
1.4 路由器及负载均衡的冗余	安井 真伸
2.1 引入反向代理……Apache 模块	伊藤 直也（Hatena）
2.2 增设缓存服务器……Squid、memcached	伊藤 直也
2.3 MySQL 同步……发生故障时的快速恢复[①]	广濑 正明（KLab）
2.4 MySQL 的 Slave + 内部负载均衡器的灵活应用示例[②]	广濑 正明
2.5 选择轻量高速的存储服务器	安井 真伸
3.1 DNS 服务器的冗余	安井 真伸
3.2 存储服务器的冗余……使用 DRBD 实现镜像	安井 真伸
3.3 网络的冗余……驱动绑定、RSTP	胜见 祐己（KLab）
3.4 引入 VLAN……使网络更加	横川 和哉（KLab）
4.1 基于 Linux 的单个主机的负载评估	伊藤 直也
4.2 Apache 的优化	伊藤 直也
4.3 MySQL 的调优诀窍[③]	广濑 正明
5.1 服务状态监控……Nagios	田中 慎司（Hatena）
5.2 服务器资源的监控……Ganglia[④]	广濑 正明
5.3 高效的服务器管理……Puppert	田中 慎司
5.4 守护进程的工作管理……Daemontools	广濑 正明
5.5 网络引导的应用……PXE、initramfs	胜见 祐己
5.6 远程维护……维护线路、Serial Console、IPMI	胜见 祐己
5.7 Web 服务器的日志处理……syslog、syslog-ng、cron、rotatelogs	胜见 祐己
6.1 Hatena 网站的内容	田中 慎司
6.2 DSAS 的内容	田中 慎司

出处

①《WEB+DB PRESS》（Vol.22）特辑 2 “MySQL 配置指导”、第 2 章 “生产环境中的同步详解”

②《WEB+DB PRESS》（Vol.38）连载 “快看！这是高手的诀窍” 可扩展的 Web 系统工房 “第 1 回：各种各样的负载均衡”

③ “5 分钟完成 MySQL 的内存关系调整！”

URL http://dsas.blog.klab.org/archives/50860867.html

④《WEB+DB PRESS》（Vol.40）连载 “快看！这是高手的诀窍” 可扩展的 Web 系统工房 “第 3 回：监控的种种”

术语整理

从网络到应用程序，本书内容涉及范围较广，其中出现了较多的术语。首先将常用的术语整理如下。

AP 服务器（Application Server）

应用服务器，即能返回动态内容的服务器。

比如 Apache + mod_perl 运行的 Web 服务器及 Tomcat 等应用程序运行的服务器。

CDN（Content Delivery Network，内容分发网络）

发送内容的网络系统。用于提高信息发送的性能和实用性。

以 Akamai 等商用服务为例，其结构上的特点是：从散布在全世界的缓存服务器中，选择离客户端较近的服务器来发送信息，据此实现性能的提升。

IPVS（IP Virtual Server，IP 虚拟服务器）

LVS（Linux Virtual Server）的成果之一，实现了负载均衡器中不可或缺的负载分流功能。

➡参考“LVS”

LVS（Linux Virtual Server，Linux 虚拟服务器）

Linux 中旨在搭建具有可扩展性的、实用性较高的系统的项目。项目成果之一即为 Linux 负载分流所设计的 IPVS。

原先为项目名，现通常作为“基于 Linux 的负载均衡器”的意思使用。

URL http://www.linuxvirtualserver.org/

NIC（Network Interface Card，网络接口卡，简称网卡）

原本是指追加网络功能所需的扩展卡。有时也作为网络接口的总称使用，不区分是扩展卡还是板载。

同时也可称为 LAN 卡、网络适配器等。

Netfilter

Linux 内核中操作网络数据包所需的协议框架。

执行分组过滤的 iptables 以及实现负载均衡的 IPVS 也应用了本 Netfilter 协议。

OSI 参考模型

用来描述数据通信网络层的模型，分为七层（Layer）框架。

以下为常见的层。

- 第七层（应用层）：HTTP 及 SMT 等通信协议
- 第四层（传输层）：TCP 及 UDP
- 第三层（网络层）：IP、ARP 及 ICMP
- 第二层（数据链路层）：以太网等

另外，像“L2 交换机”这样，有时也将“第 n 层”记为“Ln”。顺带一提，OSI 是 Open Systems Interconnection 的缩写。

VIP（Virtual IP Address，虚拟 IP 地址）

不同于物理性质的服务器及网卡，该 IP 地址会被浮动地分配某项服务或功能。

例如对于负载均衡器，接收客户端请求的 IP 地址就称为 VIP。这是因为该 IP 地址对 HTTP 等服务进行了关联，另外在冗余的 Active/Backup 架构中，唯一的 Master，即 Active 的负载均衡器也继承了该 IP 的行为。

虚拟地址通常也称为虚拟 IP 地址。

可用性（Availability）

系统停止的可能性。在可用性较高的情况下，通常该服务不会随意终止。另外，根据其字面意思，也可理解为“运行效率高”或者“1 年中的运作时间长”等。

内容（Contents）

在网络服务的环境中，内容是指返回给用户浏览器的 HTML 或图片等数据。

静态内容是指不会发生变化的内容，例如 HTML 或图片等；动态内容是指会变化的数据，根据请求的不同所返回的内容也不同。在某些情况下，动态内容并非单纯指数据本身，而是指返回动态数据的服务器站

点的程序。

服务器集群（Server Farm）

很多服务器集合而成的基础系统。根据上下文环境，有时也作为硬件设施的意思使用，与数据中心的意思相同。

在一些新闻中，有时也会形象地称为“服务器农场”。

冗余（Redundancy）

将系统的构成要素配置多个，这样即使其中一个因为发生故障而停止运作，也可以立即切换到备用设备以使服务不停止。

RAI（Redundant Arrays of Inexpensive Disks）是冗余的典型例子。

交换集线器（Switching Hub）

目前市场上几乎所有的集线器都是带有搭桥功能的交换集线器，而非“中继集线器”（Repeater Hub）。

有时也称为 L2 交换机，或者简单地称为交换机。

可扩展性（Scalability）

随着用户的增多以及规模的扩大，在某种程度上扩展系统以加强应对的能力。

横向扩展（Scale-out）

通过将内容分散到多台服务器并行处理，来提升系统整体的性能。

例如使负载均衡器下配置的 Web 服务器的数量翻倍等。

纵向扩展（Scale-up）

通过提升单个服务器的性能，来提升系统整体的性能。

例如增加服务器内存、换代到更高性能的服务器等。

准生产环境（Staging Environment）

在投入真正的服务前，进行最终的动作确认的环境（➡可参考“生产环境”）。

吞吐量（Throughput）

在网络等数据通信环境中使用，代表单位时间的传送量（➡可参考“延迟”）。

例如，虽然同样是车，但和F1赛车相比，大巴车可乘坐的人较多，因此大巴车的“吞吐量”就较大。

单点故障（Single Point of Failure）

若此处出现问题，就会令整个系统停止，即系统的要害。也叫作SPO（Single Point of Failure）。

例如，即使服务器由RAID和多路复用的电源构成，如果全部服务器都连接在同一台交换集线器上，从整个系统来看这台交换集线器即为单点故障。

数据中心（Data Center）

为了容纳服务器设备而创建的专用设备的名称。

安装有空调，并配备停电、火灾、地震等问题的应急措施，以保证每时每刻都能够正常提供服务。

守护程序（Daemon）

在后台下持续运行并发挥某种作用的程序。

例如httpd和bind等。

网段（Network Segment）

广播数据包所及范围内的网络段。虽和“冲突域”（Colision Domain）意思相近，但因为很多情况下并无冲突发生，所以很难再说“Network Segment = Colision Domain”了。

网络引导（Network Boot）

通过网络获取启动时必要的引导加载程序和内核映像并启动。

5.5节介绍的PXE是实现网络引导的方式之一。

分组（Packet）

通常指IP中数据的最小计量单位。有时也叫IP分组、IP包、数据包等。

故障转移（Failover）

在冗余系统中，在活动节点（Active Node）（服务区或者网络设备）停止时，自动通过某种行为切换到备用节点（Backup Node）。

顺带一提，如果不是自动切换，而是手动切换，通常叫作 Switch over（手动切换式故障转移）。

故障恢复（Failback）

从活动节点停止进行故障转移的状态，恢复到原始的正常状态。

帧（Frame）

以太网中数据的最小计量单位。也称为以太网帧（Ethernet Frame）。

被阻塞（Blocked）

为了等待读出或写入处理的结束而无法进行其他处理的状态，称为“因等待 I/O 而被阻塞”。

主要是针对磁盘 I/O 和网络 I/O 使用的术语，在输入输出处理时一般也会用到。

生产环境（Production Environment）

服务的运行环境（➡参考“准生产环境”）。

健康检查（Health Check）

确认检查对象的状态是否正常。

例如确认 Web 服务器是否能够响应 ping、是否能连接 TCP 的 80 端口、是否能应答 HTTP 等。通常情况下，若健康检查失败，就会向管理者发出监控对象故障的警示信息。

有时也称为“服务存活状态的监控”。

负载（Load）

“负载”的种类很多，大致可分为“CPU 负载”和“I/O 负载”。

衡量负载情况的指标通常是 load average（平均负载）这样的数值。此外 vmstat 及 top 等命令也可衡量负载。具体请参见 4.1 节。

瓶颈（Bottleneck）

阻碍系统整体性能提升的地方。

内存文件系统（Memory File System）

并非像磁盘那样永久性的存储装置，而是在内存中建立的文件系统。

虽说使用起来类似磁盘上的文件系统，但由于存储在内存中，因此

一旦重启数据就会丢失。但其拥有读写速度快等优点。

轮询（Round Robin）

对多台节点有序地派发请求。

包括 DNS 轮询和负载均衡算法等。前者是指将多个 A 记录（IP 地址）分配到一个 FQDN（完全限定域名，Fully Qualified Domain Name）上以分散请求，后者是指将请求按顺序分散到多台服务器上。

资源（Resource）

指 CPU、内存、磁盘等服务器的硬件资源。

通常说“资源被占据”就是指 CPU 使用率过高。

延迟（Latency）

在网络等数据通信领域里使用时，通常指数据投递完成所花费的时间（➡参考“吞吐量”）。

比如说，同样是车，F1 赛车就比大巴车更快速，延迟更小。

层（Layer）

➡参考“OSI 参考模型”。

负载均衡器（Load Balancer）

位于客户端与服务器之间，将客户端的请求分散到后端的多台服务器。

换句话说，就是将多台服务器合并为一台高性能的虚拟服务器的装置。

版权声明

目录