

内容分发网络 原理与实践

Principle and Practice of CDN

唐宏 陈戈 陈步华 余媛◎编著

系统地分析了CDN基本原理与关键技术
介绍了CDN的选择依据
构建了基于开源软件的CDN系统



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

内容分发网络 原理与实践

唐宏 陈戈 陈步华 余媛◎编著

**Principle and
Practice of CDN**



人民邮电出版社

北 京

图书在版编目 (C I P) 数据

内容分发网络原理与实践 / 唐宏等编著. — 北京 :
人民邮电出版社, 2018.7
ISBN 978-7-115-48803-9

I. ①内… II. ①唐… III. ①计算机网络—网络结构
IV. ①TP393.02

中国版本图书馆CIP数据核字(2018)第137137号

内 容 提 要

本书分三部分系统地分析了 CDN 基本原理与关键技术、如何选择合适的 CDN 提升业务质量、如何利用开源软件设计自有 CDN 等相关技术及方案。通过本书读者可以较为深入地了解 CDN 关键技术、CDN 市场与业务、CDN 的初步设计与开发。本书主要面向互联网技术人员、电信运营商业务与技术人员、高等院校相关专业教师与学生。

◆ 编 著 唐 宏 陈 戈 陈步华 余 媛

责任编辑 李彩珊

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>

固安县铭成印刷有限公司印刷

◆ 开本: 700×1000 1/16

印张: 14

2018 年 7 月第 1 版

字数: 297 千字

2018 年 7 月河北第 1 次印刷

定价: 79.00 元

读者服务热线: (010)81055488 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广登字 20170147 号

内容分发网络（Content Delivery Network，CDN）位于应用层之下、传输层之上的中间层，将上层业务所需要的内容通过 CDN 传送到靠近终端用户的位置，可以提升业务的服务质量。

CDN 不是一种新的网络架构，它所使用的技术也是许多成熟技术的组合，如流媒体技术、DNS 技术、HTTP 技术、IP 技术、缓存技术、IT 硬件技术等，但 CDN 的原理却不能简单地认为是这些技术的堆砌，必须把这些技术有机地结合起来，放在一些特定的业务场景下系统地分析，才能较为深入全面地了解 CDN 的关键技术与原理。

CDN 发展初期，其承载的业务单一，带宽要求不高，但近年来由于互联网与电信运营商的业务快速发展，CDN 所承载内容的内容格式、媒体协议、网络协议等都在快速变化，因此对 CDN 提出了许多新的需求，要求 CDN 能够具有兼容多种业务处理、跨网络、面向多种终端等特性，因此了解 CDN 原理也需要了解目前 IP、IT 的新技术。

CDN 除了被认为是一种网络架构、一种内容分发平台外，许多互联网公司还认为 CDN 是一种业务，它们可以向 CDN 服务提供商租用 CDN 服务，也可以通过自建 CDN 平台来为自有业务提供服务，因此作为互联网公司的技术人员，也需要了解租用与自建 CDN 的对比。

本书分三部分系统地分析了 CDN 基本原理与关键技术、如何选择合适的 CDN 提升业务质量、如何利用开源软件设计自有 CDN 等相关技术及方案。

本书的作者均是中国电信具有多年 CDN 研发经验的技术人员，对 CDN 的研发、测试、

运营都有较为深刻的认识，在 CDN 领域取得了许多专利、标准等成果。唐宏负责本书的总体架构设计、关键技术、新技术部分的编写；陈戈负责 CDN 租用、开源 CDN 架构设计部分的编写；陈步华负责 CDN 产业、CDN 选择、CDN 组网部分的编写；余媛负责开源 CDN 部分的编写。另外，梁洁、庄一嵘对本书内容进行了校对、检查并提出了许多建议，黄宇民对书中的部分技术细节进行了修订，卢琳、何道琼对本书的插图与文本进行了编辑，对此深表谢意。

编 者

2018 年 3 月

第一部分 CDN 基本原理与关键技术

第 1 章 影响互联网应用质量的关键	3
1.1 互联网应用发展	3
1.2 互联网应用质量	4
1.2.1 质量是互联网应用的生命	4
1.2.2 网络性能是影响质量的关键	7
1.3 提高互联网应用质量的方法	10
1.3.1 集中式部署带来的问题	10
1.3.2 利用网站镜像加速	11
1.3.3 利用 CDN 进行加速	12
第 2 章 CDN 基本原理	16
2.1 CDN 的基本概念	16
2.1.1 CDN 的定义	16
2.1.2 CDN 可承载的内容	17
2.2 CDN 的工作过程	18
2.2.1 CDN 的基本工作过程	18
2.2.2 CDN 内容接入	20
2.2.3 CDN 用户请求调度	21
2.2.4 CDN 内容分发	22

2.2.5	CDN 内容服务	23
第 3 章	典型的 CDN 架构与组网	25
3.1	CDN 功能平面	25
3.2	CDN 内部网元	26
3.3	CDN 部署架构	31
3.4	CDN 间组网	33
第 4 章	CDN 关键技术	35
4.1	统一内容 ID	36
4.1.1	统一资源定位符	37
4.1.2	CDN 内容统一 ID	37
4.2	本地负载均衡	38
4.2.1	负载均衡技术	38
4.2.2	负载均衡的技术分类	39
4.2.3	负载均衡的算法	41
4.3	用户请求路由调度/全局负载均衡	43
4.3.1	基于 DNS 的用户调度	44
4.3.2	基于 HTTP 的重定向	47
4.4	内容缓存技术	49
4.4.1	缓存技术	49
4.4.2	缓存替换算法	50
4.5	流媒体技术	55
4.5.1	实时流媒体技术: RTSP	55
4.5.2	渐进式下载流媒体技术: HTTP Streaming	58
4.5.3	自适应流媒体技术	62
4.6	服务鉴权技术	69
4.6.1	常见的服务鉴权技术	69
4.6.2	CDN 服务鉴权机制	71
第 5 章	CDN 新技术	73
5.1	前端优化技术	73
5.2	动态加速技术	75

5.3 SDN 调度技术	79
5.4 P2P 流媒体技术	81
5.5 应用协议加速技术	82
5.6 智能协同技术	83
5.7 NVMe 技术	84

第二部分 CDN 的选择

第 6 章 CDN 产业与市场发展	89
6.1 CDN 产业发展	89
6.1.1 CDN 产业的发展历程	89
6.1.2 CDN 服务提供商类型	91
6.1.3 CDN 市场的发展特点	92
6.2 CDN 发展趋势	93
6.2.1 CDN 业务发展趋势	93
6.2.2 CDN 市场发展趋势	95
6.2.3 CDN 网络发展趋势	98
6.3 CDN 服务商面临的挑战	102
6.3.1 不可忽视的安全因素	103
6.3.2 大数据流量的利用	103
6.3.3 推进统一的技术标准	103
6.3.4 定制化的技术创新要求	104
第 7 章 租用 CDN 与自建 CDN 的选择	105
7.1 租用 CDN 与自建 CDN 对比	105
7.1.1 业务需求	105
7.1.2 CDN 与成本分析	106
7.1.3 CDN 租用与自建结合	106
7.2 租用 CDN 的选择	107
7.2.1 CDN 服务类型与功能	108
7.2.2 CDN 容量与分布	109
7.2.3 CDN 运营维护	109
7.2.4 CDN 价格	110

7.2.5 多 CDN 租用	111
7.3 自建 CDN 的选择	112
7.3.1 开源 CDN 软件与商业 CDN 软件对比	112
7.3.2 业务需求变化	112
7.3.3 开发和维护能力	113
第 8 章 租用 CDN 实施的考虑与评估	114
8.1 CDN 需求确认	115
8.2 CDN 测试验证	117
8.2.1 CDN 服务切换	117
8.2.2 CDN 服务质量测试	118
8.3 CDN 服务监控	122
8.4 多 CDN 租用调度	124
8.4.1 多 CDN 租用调度系统 (TMS)	125
8.4.2 服务调度流程	127
第 9 章 自建 CDN 实施的考虑与评估	129
9.1 自建 CDN 案例分析	129
9.2 自建 CDN 中的开源软件	132
9.2.1 缓存系统	133
9.2.2 内容管理系统	133
9.2.3 内容路由系统	134
9.2.4 监控系统	134
9.3 自建 CDN 部署	135
9.4 自建 CDN 技术指标	136
9.4.1 命中率	136
9.4.2 吞吐量	138
9.4.3 并发值	139
9.4.4 响应时间	140
9.4.5 MDI	141
9.4.6 MOS	142
9.4.7 稳定性和可靠性	143

第三部分 基于开源的自建 CDN 设计

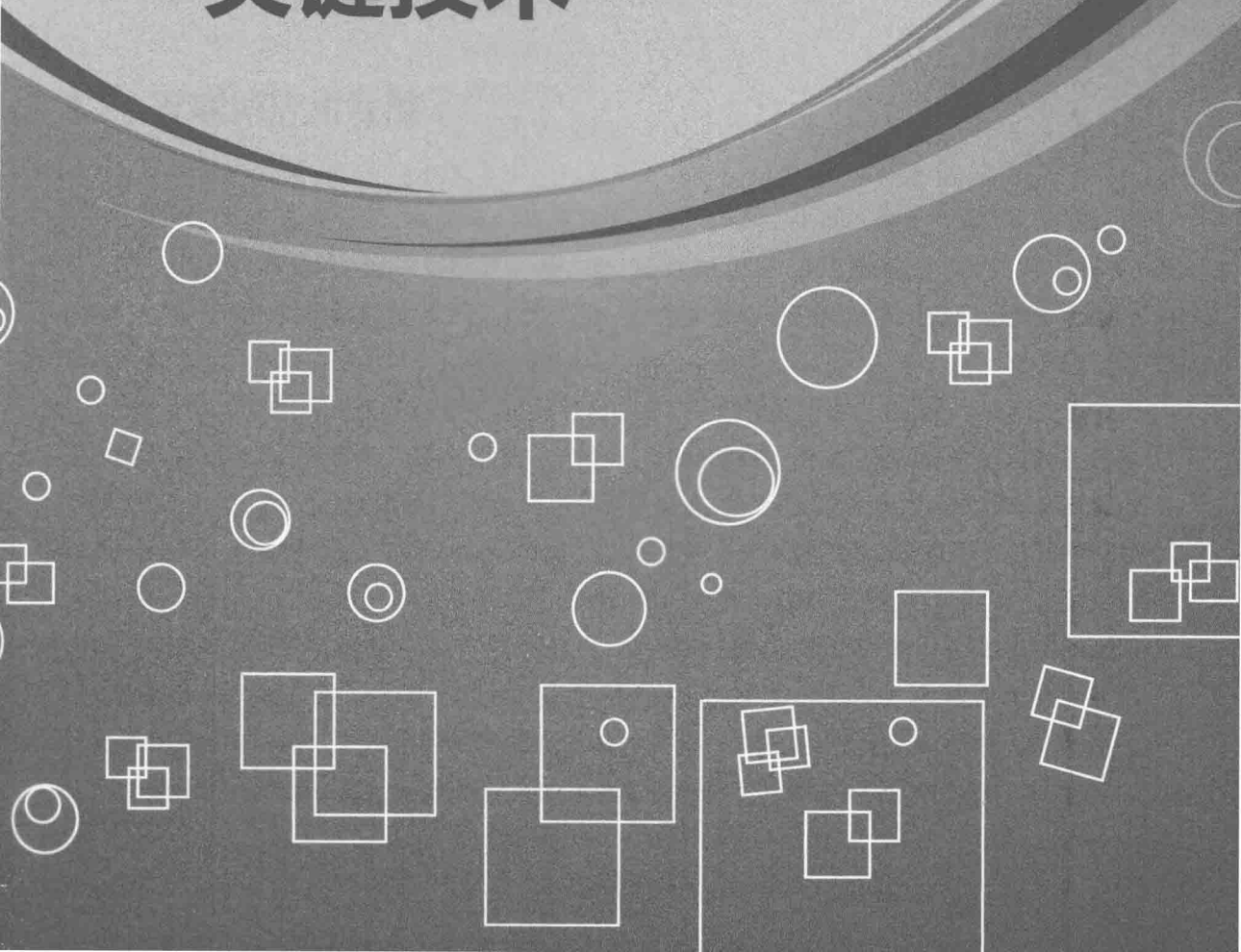
第 10 章 开源 CDN 架构设计	147
10.1 业务需求	147
10.2 开源 CDN 总体架构	149
10.3 CDN 网络规划	150
第 11 章 流服务缓存节点模块的设计	152
11.1 流服务缓存节点的特性	152
11.2 流服务缓存节点开源软件简介	153
11.2.1 Squid	153
11.2.2 Quagga	153
11.2.3 LVS	154
11.2.4 Keepalived	155
11.2.5 Nginx	155
11.2.6 Lua	155
11.3 模块设计	156
11.3.1 代理缓存 (Squid)	156
11.3.2 四层负载均衡 (OSPF+LVS+Keepalived)	157
11.3.3 七层负载均衡 (Nginx+Lua)	158
11.4 环境配置	158
11.4.1 Squid 安装与配置	158
11.4.2 OSPF 安装与配置	161
11.4.3 LVS 安装	162
11.4.4 Keepalived 安装与配置	163
11.4.5 Nginx 安装与配置	165
11.4.6 lua-nginx-module 模块安装	166
第 12 章 内容库模块的设计	168
12.1 内容库的特性	168
12.2 内容库开源软件简介	168
12.2.1 FTP	168

12.2.2	Ceph	169
12.3	模块设计	170
12.3.1	内容注入 (FTP)	171
12.3.2	分布式内容存储 (Ceph)	171
12.3.3	内容分发 (Nginx)	171
12.4	环境配置	172
12.4.1	FTP 服务器搭建	172
12.4.2	Ceph 安装与配置	173
12.4.3	Nginx 安装与配置	175
第 13 章	全局用户请求调度模块的设计	176
13.1	基于 DNS 的流量管理服务设计	176
13.1.1	基于 DNS 流量管理的开源软件简介	176
13.1.2	模块设计	177
13.1.3	环境配置	177
13.2	基于 HTTP 的应用层调度服务设计	180
13.2.1	基于 HTTP 调度的开源软件简介	180
13.2.2	模块设计	180
13.2.3	环境配置	181
第 14 章	网络管理模块的设计	182
14.1	网络管理工作流程	182
14.2	网络管理开源软件简介	183
14.2.1	Zabbix	183
14.2.2	InfluxDB	185
14.2.3	Grafana	187
14.3	模块设计	188
14.3.1	数据采集 (Zabbix)	188
14.3.2	数据存储 (InfluxDB)	189
14.3.3	数据展示 (Grafana)	189
14.4	环境配置	189
14.4.1	Zabbix 安装与配置	189
14.4.2	InfluxDB 安装与配置	191

14.4.3 Grafana 安装与配置	194
第 15 章 基于开源的自建 CDN 测试验证	195
15.1 系统测试	195
15.1.1 测试目的	195
15.1.2 测试方法	196
15.1.3 测试拓拓扑组网	197
15.1.4 测试内容	198
15.1.5 测试过程	199
15.2 现网测试	205
15.2.1 测试目的	205
15.2.2 测试方法	206
15.2.3 测试过程	207
参考文献	211

第一部分

CDN基本原理与 关键技术



影响互联网应用质量的关键

1.1 互联网应用发展

互联网已经高速发展了 30 多年，从各方面改变了人类的生活，是人类最伟大的发明之一。最初，互联网只是单纯用来进行数据通信的，直到 20 世纪 80 年代末，万维网（即 WWW）出现，使互联网的架构格局发生了巨大的变化，并使其成为了在全球范围内能够实现资源共享和数据传输的分布式网络。

最开始，互联网应用仅仅是一些利用 HTML 等工具制作的网站内容集合。然而，随着互联网的发展，除了网站以外，还产生了电子商务、社交网络、游戏、定位导航等各种各样的互联网应用，但网站业务依然是互联网应用的最主要应用。网站业务作为人们获取新闻要点和第一手资讯的主要来源，已从传统单一的文字形式，飞速发展到音频、视频以及弹幕、增强现实（Augmented Reality，AR）等实时互动、参与感更加真实强烈的信息传递方式。

近年来，除了网站业务以外，网络游戏、流媒体视频、电子商务、社交网络、地图导航等业务也飞速发展，尤其是在移动互联网技术的大规模部署下，各种互联网应用已呈现爆发式的增长，互联网内容的组织也越来越复杂，如图 1-1 所示。互联网内容都是由网页、音频、视频等构成的，这些内容的发展对互联网应用质量有着很大影响。



图 1-1 互联网内容的爆炸式增长示意

互联网应用的内容数量和类型不断快速增长，从最初的页面只包含 2~3 个对象到现在包含众多对象，这就意味着服务器的往返交互次数会随着内容的复杂程度而快速增加，应用加载所需要的时间也会越长。

1.2 互联网应用质量

1.2.1 质量是互联网应用的生命

(1) 用户对互联网应用质量的期望

一项研究用户互联网行为的实验结果显示，用户在 4 s 以内看到网站的页面加载完成或者开始产生页面渲染，会让用户产生非常满意的访问体验，这就是互联网行业里著名的“4 s 法则”。当用户向网站发起访问请求时，若超过 4 s，网页还没有被加载下来，30%的用户会选择不再等待，并关掉页面。对于用户来说，6~8 s 的页面加载时间已经是能够忍耐的极限了，因此，8 s 成为了一个临界值。如果网站的加载速度总是超出 8 s，用户会觉得网站的访问体验很

差。假如用户对这个网页有必不可少的访问需求，最多还会等待到 12 s 左右，要是网页依然没有被加载完成，该网站会产生大量用户的流失。上述实验结果见表 1-1，由于用户倾向性地认为网站打开速度越快，其服务质量就越高，因此，用户对网站的响应速度有着很高的要求。

表 1-1 用户访问体验与响应时间的关系

响应时间	用户感觉
0~4 s	很快
4~6 s	有一点点慢
6~8 s	机器在工作
8~12 s	先干点别的
>12 s	不能用了

当用户向网站发起访问请求，而且在很长一段时间内没有获得响应时，用户的忍受程度会随着网站打开时间的变长而降低，甚至会质疑网站本身的可靠性和质量，认为网站服务器存在一些故障。并且，网站缓慢的打开速度可能会让用户忘记原本访问网站的目的，不得不重新回忆下一步要做什么，从而进一步导致用户的访问体验恶化。

速度和性能是两个相对的概念。由于商业模式、应用场景以及功能复杂性等各不相同，每个网站业务都要满足其特定的用户需求。从图 1-2 用户访问电商网站感觉缓慢的后果中可以看出，所有网站都必须在一定时间内对用户做出响应。因此，从一定程度上看，网站速度是用户满意程度和网站性能的关键因素。

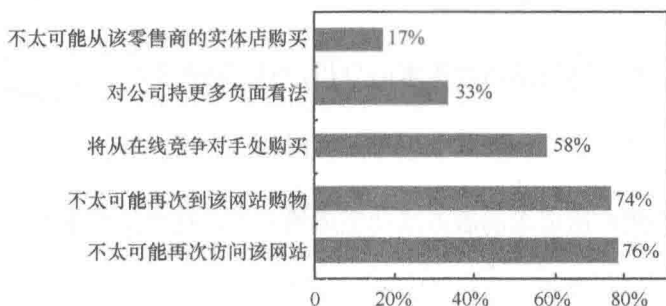


图 1-2 用户访问电商网站感觉缓慢的后果