

国外计算机科学教材系列

分布式操作系统

Distributed Operating Systems

[美] Andrew S. Tanenbaum 著

陆丽娜 伍卫国 刘隆国 等译校



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
URL: <http://www.phei.com.cn>



PRENTICE HALL 出版公司

国外计算机科学教材系列

分布式操作系统

Distributed Operating Systems

Andrew S. Tanenbaum · 著

陆丽娜 伍卫国 刘隆国 等译校

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 提 要

分布式操作系统是为分布式计算机系统配置的一种操作系统。本书涵盖了分布式操作系统的所有内容,并重点介绍设计和构造分布式操作系统的基本原理、算法和实现技术。全书共 11 章,前 6 章较全面地介绍了分布式系统环境中的通信、同步、进程、文件系统和存储器管理等方面的概念与算法;后 4 章详细地给出了 4 个分布式系统的实例: Amoeba、Mach、Chorus 和 DCE,介绍其设计思想和实现技术。本书既注重基础知识的系统性,同时兼顾选材的先进性,内容全面经典、层次清楚。

本书可作为学习、使用和讲授分布式操作系统的大学生、研究生和教师的教学用书,也可供从事分布式计算机系统的科技工作者阅读和参考。

©1995 by Prentice-Hall, Inc.

本书中文简体版由电子工业出版社和美国 Prentice-Hall 出版公司合作出版。未经许可,不得以任何手段和形式复制或抄袭本书内容。版权所有,侵权必究。

图书在版编目(CIP)数据

分布式操作系统/(美) 坦恩鲍姆(Tanenbaum.A.S.)著;

陆丽娜等译 - 北京: 电子工业出版社, 1999.12

(国外计算机科学教材系列)ISBN 7 - 5053 - 5487 - 6

I.分... II.①坦... ②陆... III.分布式操作系统-教材 IV.TP316

中国版本图书馆 CIP 数据核字(1999)第 63451 号

从 书 名: 国外计算机科学教材系列

书 名: 分布式操作系统

原 书 名: **Distributed Operating Systems**

著 者: [美] Andrew S.Tanenbaum

译 校 者: 陆丽娜 伍卫国 刘隆国 等

责任编辑: 陆伯雄

特约编辑: 杨宝珍

印 刷 者: 北京天竺颖华印刷厂

出版发行: 电子工业出版社 URL:<http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 28.5 字数: 672 千字

版 次: 1999 年 12 月第 1 版 1999 年 12 月第 1 次印刷

印 数: 8000 册

定 价: 45.00 元

书 号: ISBN 7-5053-5487-6/TP·2770

著作权合同登记号 图字: 01-1999-2973

凡购买电子工业出版社的图书,如有缺页、倒页、脱页者,请向购买书店调换。

若书店售缺,请与本社发行部联系调换。电话 68279077

出版说明

计算机科学的迅速发展是 20 世纪科学发展史上最伟大的事件之一。从 1946 年第一台笨重而体积庞大的计算机的发明至今, 仅仅半个多世纪, 计算机已经变得小巧无比却又能力非凡。它的应用已经渗透到了社会的各个方面, 成为当今所谓的信息社会的最显著的特征。

处于世纪之交科技进步的大潮中, 我国正在加强计算机科学的高等教育, 着眼于为下一世纪培养高素质的计算机人才, 以适应信息社会加速度发展的需要。当前, 全国各类高等院校已经或计划在各专业基础课程规划中增加计算机科学的课程内容, 而作为与计算机科学密切相关的计算机、通信、信息等专业, 更是在酝酿着教学的全面革新, 以期规划出一整套面向 21 世纪的、具有中国高校计算机教育特色的课程计划和教材体系。值此, 我们不妨借鉴并引进国外具有先进性、实用性和权威性的大学计算机教材, 洋为中用, 以更好地服务于国内的高校教育。

美国 Prentice Hall 出版公司是享誉世界的高校教材出版商, 自 1913 年公司成立以来, 即致力于教育图书的出版。它所出版的计算机教材在美国为众多大学所采用, 其中有不少是专业领域中的经典名著。许多蜚声世界的教授学者成为该公司的资深作者, 如: 道格拉斯·科默(Douglas Comer), 安德鲁·坦尼伯姆(Andrew Tanenbaum), 威廉·斯大林(William Stallings)……几十年来, 他们的著作教育了一批批不同肤色的莘莘学子, 使这些教材同时也成为全人类的共同财富。

为了保证本系列教材翻译出版的质量, 电子工业出版社和 Prentice Hall 出版公司共同约请北京地区的清华大学、北京大学、北京航空航天大学, 上海地区的上海交通大学、复旦大学, 南京地区的南京大学、解放军通信工程学院等全国著名的高等院校的教学第一线的几十位教师参加翻译工作。这中间有正在讲授同类教材的年轻教师和博士, 有积累了几十年教学经验的教授和博士生导师, 还有我国著名的计算机科学家。他们的辛勤劳动保证了本系列丛书得以高质量地出版面世。

如此大规模地引进计算机科学系列教材, 在我们还是第一次。除缺乏经验之外, 还由于我们对计算机科学的发展, 对中国高校计算机教育特点认识的不足, 致使在选题确定、翻译、出版等工作中, 肯定存在许多遗憾和不足之处, 恳请广大师生和其他读者提出批评、建议。

电子工业出版社

URL:<http://www.phei.com.cn>

Prentice Hall 出版公司

URL:<http://www.prenhall.com>

译者序

近年来,分布式处理系统获得了突飞猛进的发展,并呈现出前所未有的广阔前景。为此我们必须学习和系统了解当前国内外分布式系统最新的理论与技术。坦尼鲍姆教授是国际知名的计算机科学和教育家。他在计算机操作系统、分布式操作系统、计算机网络领域都有很深的造诣。对操作系统的介绍,他先后完成了面向大学生和研究生的三卷专著:《操作系统:设计和实现》、《分布式操作系统》和《现代操作系统》,这些教材被国内外的许多大学广为采用。为此我们翻译了《分布式操作系统》一书。

《分布式操作系统》一书涵盖了分布式操作系统的所有的内容。较全面地介绍了分布式系统环境中的通信、同步、进程、文件系统和存储器管理等方面的概念与算法,并详细地给出了分布式系统设计实例,它不仅可作为一本很好的教材,同时对分布式系统的设计将带来很大帮助。

本书的特点是理论与实践紧密结合,第1章到第6章讲述原理,第7章到第10章分别介绍4个分布式操作系统实例。通过学习,使读者对分布式操作系统的原理和实现有一个完整的认识 and 了解,并能够了解与跟踪当前的最新技术。

本书的第1章由陆丽娜翻译,第2、3章由刘归荣翻译,第4、5、6章由吴昊、艾小平、翟刚翻译,第7、8、9、10、11章由刘隆国、伍卫国翻译。全书由陆丽娜教授审校并统稿。书中所有的图表由张兴军、冯锐与邢飞翻译并绘制。

在本书翻译过程中,部分译稿在西安交通大学电子信息工程学院计算机系97级研究生分布式系统课程中进行了试用,许多学生提出了宝贵的意见。伊景冰对本书的文字修饰提出了宝贵意见,在此对他们表示衷心的感谢。

虽然在翻译过程中我们尽量做到尊重原意、翻译准确,希望它能对国内的分布式系统的教学与科研有所帮助,但由于水平所限,不当和疏漏之处在所难免,敬请读者提出宝贵意见。

译者
西安交通大学电子与信息学院
计算机科学与技术系
1999.5

序 言

在 *Distributed Operating Systems* 一书出版后, 我已经完成了对操作系统介绍的三步曲。这个三步曲中的三本书是:

- *Operating Systems: Design and Implementation* (操作系统: 设计和实现)
- *Distributed Operating Systems* (分布式操作系统)
- *Modern Operating Systems* (现代操作系统)

然而, 这三本书并不是完全独立的。对于有两门操作系统课程(或一门本科操作系统课程和一门研究生操作系统课程)的学校, 一种可能的选择可以是使用 *Operating Systems: Design and Implementation* 作为第一门课程, 而用 *Distributed Operating Systems* 作为第二门课程。

第一本书介绍单处理机系统中操作系统的一些基本原理, 包括进程、同步、I/O、死锁、内存管理、文件系统、安全性等等。它还应用 MINIX 系统对这些原理进行十分详细的说明, MINIX 系统是 UNIX 系统的克隆体, 它的源文件在书的附录中给出。MINIX 光盘附在书后, 可在 IBM PC (8088 和更高档的), Atari、Amiga、Macintosh 和 SPARC 处理机上运行。

第二本书(即本书), 详细介绍了分布式操作系统, 包括分布式系统环境中的通信、同步、进程、文件系统和内存管理等方面。并详细地给出了 4 个分布式系统的例子: Amoeba、Mach、Chorus 和 DCE。Amoeba 系统对于大学中用于教学方面是免费的。它可在 Intel 386/486、SPARC 和 Sun3 处理机上运行。要想知道怎样得到 Amoeba 系统的信息, 请从 <ftp.cs.vu.nl> 上 FTP 下载文件 `amoeba/Intro.ps.Z` 或者与作者通过电子邮件联系, 地址是: ast@cs.vu.nl。首先要提醒一下各个用户, Amoeba 要比 MINIX 复杂得多: 光文档(可通过 FTP 获得)就有 1000 多页, 并且系统要想运行得较好, 至少需要 5 个大型机和一个以太网环境。

通过先后学习这两本书并使用 MINIX 和 Amoeba 系统, 学生能够对单处理机操作系统和分布式操作系统的原理和实现有一个全面的认识 and 了解。现在这三本书都已完成, 我计划要修改 MINIX 操作系统及介绍它的书籍。

对于时间不宽松的大学和计算机专家来说, *Modern Operating Systems* 一书是其它两本书的一个浓缩。那两本书介绍了单处理机操作系统和分布式操作系统的基本原理, 但是没有 MINIX 这样详细的例子。它还省略了这本书中介绍的许多高级的主题, 包括 ATM 的介绍, 容错分布式系统、实时分布式系统、分布式共享存储器, Chorus 和 DCE 等等。总的来说, *Modern Operating Systems* 一书中大约省略了关于分布式操作系统的 230 页的内容。

许多人帮我完成了此书的创作。我特别感谢以下朋友, 他们都阅读了原稿的一些部分并给我提出了许多宝贵的改进意见, 他们是: Irina Athanasiu、Henri Bal、Saniya Ben Hassen、David Black、John Carter、Randall Dean、Wiebren de Jonge、John Dugas、Dick

Grune、Anoop Gupta、Frans Daashoek、Marcus Koebler、Hermann Kopetz、Ed Lazowska、Dan Lenoski、Kai Li、Marc Maathuis、David Mosberger、Douglas Orr、Craig Partridge、Carlton Pu、Marc Rozier、Rich Salz、Mike Schroeder、Karsten Schwan、Greg Sharp、Dennis Shasha、Sol Shatz、Jennifer Steiner、Chuck Thacker、John Turek、Walt Tuvell、Leendert van Doorn、Robbert van Renesse、Kees Verstoep、Ellen Zegura、Willy Zwaenpoel 和一些无名的审阅者。我的编辑者 Bill Zobrist，承受着为将本书写得尽善尽美所带来的麻烦，而毫无怨言。

虽然有这么多人的协助，但错误总还是存在的。不管多少人阅读了原稿，仍有错误存在是不可避免的。如果您发现了错误，请通过电子邮件和我联系。

最后，我要再一次感谢 Suzanne。在帮我写完 8 本书后，她又帮我写另一本书了，她的耐心和爱是无穷的。我同样要感谢 Barbara 和 Marvin 两人让我单独使用他们的计算机。教我们如何使用 PC 字处理程序，使我更加喜欢 UNIX 中的 troff 排版编辑程序了。最后，我要感谢 Little Bram 在我写书的时候，为我提供一个很安静的环境。

Andrew S. Tanenbaum

(使用本书中文版作为教材授课的教师，请联络 Prentice Hall 公司北京代表处，电子邮件地址为：ssbj@bupt.edu.cn，通信地址为：北京西三环北路 19 号外研社大厦 2205 室，邮编 100081。目前教师指导书仅能免费提供英文版。)

目 录

第 1 章 分布式系统概述.....	1
1.1 什么是分布式系统?	1
1.2 目标.....	2
1.2.1 分布式系统与集中式系统相比较而言的优点	2
1.2.2 分布式系统与独立 PC 机相比较的优点	4
1.2.3 分布式系统的缺点	5
1.3 硬件概念	6
1.3.1 基于总线的多处理机.....	8
1.3.2 交换型多处理机	9
1.3.3 基于总线的多计算机.....	10
1.3.4 交换型多计算机	11
1.4 软件概念	11
1.4.1 网络操作系统	12
1.4.2 真正的分布式系统	14
1.4.3 多处理机分时系统	15
1.5 设计中的问题.....	17
1.5.1 透明性(Transparency)	17
1.5.2 灵活性	19
1.5.3 可靠性	20
1.5.4 性能.....	21
1.5.5 可伸缩性.....	22
1.6 小 结.....	23
习 题.....	24
第 2 章 分布式系统的通信	25
2.1 分层协议	25
2.1.1 物理层	27
2.1.2 数据链路层	28
2.1.3 网络层	29
2.1.4 传输层	29
2.1.5 会话层	30
2.1.6 表示层	30
2.1.7 应用层	30
2.2 异步传输模式网(ATM 网).....	30
2.2.1 什么是异步传输模式.....	30
2.2.2 ATM 物理层	32
2.2.3 ATM 层	32

2.2.4	ATM 适配层	33
2.2.5	ATM 交换	34
2.2.6	ATM 对分布式系统的影响	35
2.3	客户-服务器模式	36
2.3.1	客户机和服务器	36
2.3.2	客户和服务器的一个示例	37
2.3.3	寻址	41
2.3.4	阻塞与非阻塞原语	43
2.3.5	有缓冲和无缓冲原语	45
2.3.6	可靠的和非可靠原语	46
2.3.7	客户-服务器模式的实现	47
2.4	远程过程调用	49
2.4.1	基本 RPC 操作	50
2.4.2	参数传递	53
2.4.3	动态捆绑	56
2.4.4	失败情况下的 RPC 语义	58
2.4.5	实现的问题	62
2.4.6	问题领域	70
2.5	组通信	72
2.5.1	组通信的引入(Introduction to Group Communication)	72
2.5.2	设计的问题	73
2.5.3	在 ISIS(组合软件调用系统)中的组通信	80
2.6	小结	83
	习 题	84
第 3 章	分布式系统的同步	86
3.1	时钟同步	86
3.1.1	逻辑时钟	87
3.1.2	物理时钟	90
3.1.3	时钟同步算法	92
3.1.4	使用同步时钟	96
3.2	互斥	97
3.2.1	集中式算法	97
3.2.2	分布式算法	98
3.2.3	令牌环算法	100
3.2.4	三种算法的比较	101
3.3	选举算法	102
3.3.1	欺负 (Bully) 算法	103
3.3.2	环算法	104
3.4	原子事务	105
3.4.1	原子事务简介	105
3.4.2	事务模型	106

3.4.3	实现.....	110
3.4.4	并发控制.....	113
3.5	分布式系统中的死锁.....	116
3.5.1	分布式死锁检测.....	117
3.5.2	分布式死锁预防.....	120
3.6	小结.....	122
习 题		122
第 4 章	分布式系统中的进程和处理机.....	124
4.1	线程.....	124
4.1.1	线程简介.....	124
4.1.2	线程的用途.....	126
4.1.3	线程包的设计问题.....	128
4.1.4	实现一个线程包.....	130
4.1.5	线程和远程过程调用 (RPC).....	134
4.2	系统模型.....	136
4.2.1	工作站模型.....	136
4.2.2	使用空闲工作站.....	138
4.2.3	处理机池模型.....	141
4.2.4	混合模型.....	144
4.3	处理机分配.....	144
4.3.1	分配模型.....	144
4.3.2	处理机分配算法的设计问题.....	146
4.3.3	处理机分配算法的实现问题.....	148
4.3.4	处理机分配算法举例.....	149
4.4	分布式系统的调度.....	154
4.5	容错.....	155
4.5.1	组成部件错误.....	156
4.5.2	系统失效.....	156
4.5.3	同步系统与异步系统.....	157
4.5.4	使用冗余.....	157
4.5.5	使用主动复制方法的容错.....	158
4.5.6	使用主机后备的容错.....	159
4.5.7	容错系统中的协同一致.....	161
4.6	实时分布式系统.....	163
4.6.1	什么是实时系统?.....	163
4.6.2	设计问题.....	165
4.6.3	实时通信.....	168
4.6.4	实时调度.....	171
4.7	小结.....	176
习 题		176
第 5 章	分布式文件系统.....	179

5.1	分布式文件系统设计	179
5.1.1	文件服务接口	179
5.1.2	目录服务器接口	181
5.1.3	文件共享的语义	185
5.2	分布式文件系统的实现	187
5.2.1	文件的使用	188
5.2.2	系统结构	189
5.2.3	高速缓存(caching, 超缓存)	192
5.2.4	复制	196
5.2.5	例子: SUN 公司的网络文件系统	199
5.2.6	学到的教训	204
5.3	分布式文件系统的发展趋势	205
5.3.1	新的硬件	205
5.3.2	规模	207
5.3.3	广域网	207
5.3.4	移动用户	208
5.3.5	容错	209
5.3.6	多媒体	209
5.4	小结	209
习 题		210
第 6 章	分布式共享存储器	212
6.1	简介	213
6.2	什么是共享存储器?	214
6.2.1	芯片存储器	214
6.2.2	基于总线的多处理机	214
6.2.3	基于环的多处理机	218
6.2.4	交换式多处理机	220
6.2.5	NUMA 多处理机	224
6.2.6	分布式共享系统的比较	227
6.3	一致性模型	229
6.3.1	严格一致性 (Strict Consistency)	230
6.3.2	顺序一致性(Sequential Consistency)	231
6.3.3	因果一致性(Causal Consistency)	234
6.3.4	PRAM 一致性(PRAM Consistency)和处理器一致性(Processor Consistency)	235
6.3.5	弱一致性 (Weak Consistency)	236
6.3.6	释放一致性 (Release Consistency)	238
6.3.7	入口一致性 (Entry Consistency)	240
6.3.8	一致性模型总结	241
6.4	基于分页的分布式共享存储器	242
6.4.1	基本设计	242
6.4.2	复制	243

6.4.3	粒度 (Granularity)	244
6.4.4	实现顺序一致性 (Achieving Sequential Consistency)	245
6.4.5	寻找拥有者 (Finding The Owner)	247
6.4.6	寻找拷贝	248
6.4.7	页面置换 (Page Replacement)	249
6.4.8	同步	250
6.5	共享变量的分布式共享存储器	250
6.5.1	Munin	251
6.5.2	Midway	256
6.6	基于对象的分布共享内存	257
6.6.1	对象	258
6.6.2	Linda	259
6.6.3	Orca	264
6.7	比较	268
6.8	小结	269
	习 题	270
第 7 章	实例研究 1: Amoeba	272
7.1	Amoeba 介绍	272
7.1.1	Amoeba 的发展史	272
7.1.2	研究目标	272
7.1.3	Amoeba 的系统结构	273
7.1.4	Amoeba 微内核	275
7.1.5	Amoeba 服务器	276
7.2	Amoeba 中的对象 (object) 和权能 (capabilities)	277
7.2.1	权能 (Capability)	277
7.2.2	对象保护	278
7.2.3	标准操作	279
7.3	Amoeba 中的进程管理	280
7.3.1	进程	280
7.3.2	线程	282
7.4	Amoeba 中的内存管理	283
7.4.1	段 (segment)	283
7.4.2	段映射 (mapped segments)	283
7.5	Amoeba 的通信	284
7.5.1	远程过程调用(RPC)	284
7.5.2	Amoeba 的组通信	287
7.5.3	快速本地互联网协议 FLIP (The Fast Local Internet Protocol)	293
7.6	Amoeba 服务器	299
7.6.1	子弹服务器 (Bullet Server)	299
7.6.2	目录服务器	302
7.6.3	复制服务器	306

7.6.4	运行服务器	306
7.6.5	引导服务器 (the boot server)	308
7.6.6	TCP/IP 服务器	308
7.6.7	其他服务器	308
7.7	小结	308
习 题		309
第 8 章	实例研究 2: Mach	311
8.1	有关 Mach 的介绍	311
8.1.1	Mach 的发展历史	311
8.1.2	Mach 的设计目标	312
8.1.3	Mach 微内核	312
8.1.4	Mach 的 BSD UNIX 服务器	314
8.2	Mach 中的进程管理	314
8.2.1	进程	314
8.2.2	线程	316
8.2.3	调度	319
8.3	Mach 的存储管理	321
8.3.1	虚拟内存	322
8.3.2	存储共享	324
8.3.3	外部存储管理器	326
8.3.4	Mach 中的分布式共享存储	329
8.4	Mach 中的通信	330
8.4.1	端口	330
8.4.2	消息的发送与接收	334
8.4.3	网络消息服务器	338
8.5	Mach 的 UNIX 仿真	340
8.6	小结	341
习 题		342
第 9 章	实例研究 3: Chorus	343
9.1	Chorus 简介	343
9.1.1	Chorus 的发展史	343
9.1.2	Chorus 的设计目标	344
9.1.3	Chorus 系统结构	344
9.1.4	内核概念	346
9.1.5	内核结构	348
9.1.6	UNIX 子系统	349
9.1.7	面向对象子系统	349
9.2	Chorus 中的进程管理	349
9.2.1	进程	349
9.2.2	线程	350

9.2.3	调度.....	351
9.2.4	陷阱、异常和中断.....	352
9.2.5	进程管理的内核调用.....	353
9.3	Chorus 的内存管理.....	354
9.3.1	区域和段.....	354
9.3.2	映像程序 (Mapper)	355
9.3.3	分布式共享存储器.....	356
9.3.4	内存管理的内核调用.....	356
9.4	Chorus 中的通信.....	358
9.4.1	消息.....	358
9.4.2	端口.....	358
9.4.3	通信操作.....	359
9.4.4	通信的内核调用.....	360
9.5	Chorus 中的 UNIX 仿真.....	361
9.5.1	UNIX 进程的结构.....	362
9.5.2	对 UNIX 的扩展.....	362
9.5.3	Chorus 上 UNIX 的实现.....	363
9.6	COOL: 一个面向对象的子系统.....	367
9.6.1	COOL 的体系结构.....	367
9.6.2	COOL 基层.....	368
9.6.3	COOL 通用运行时系统.....	368
9.6.4	语言运行时系统.....	369
9.6.5	COOL 的实现.....	369
9.7	Amoeba、Mach 和 Chorus 的比较.....	369
9.7.1	指导思想.....	369
9.7.2	对象.....	371
9.7.3	进程.....	371
9.7.4	内存模式.....	372
9.7.5	通信.....	372
9.7.6	服务器.....	373
9.8	小结.....	374
习题		375
第 10 章	实例研究 4: DCE	376
10.1	关于 DCE 的介绍.....	376
10.1.1	DCE 的历史.....	376
10.1.2	DCE 的目标.....	376
10.1.3	DCE 部件.....	377
10.1.4	信元.....	379
10.2	线程.....	381
10.2.1	DCE 线程介绍.....	381
10.2.2	调度.....	382

10.2.3	同步	383
10.2.4	线程调用	384
10.3	远程过程调用	387
10.3.1	DCE RPC 的目标	387
10.3.2	客户与服务器的编写	387
10.3.3	客户到服务器的绑定	389
10.3.4	RPC 的执行	390
10.4	时间服务	390
10.4.1	DTS 时间模型	391
10.4.2	DTS 实现	392
10.5	目录服务	394
10.5.1	名字	394
10.5.2	信元目录服务	395
10.5.3	全局目录服务	398
10.6	安全服务	401
10.6.1	安全模式	402
10.6.2	安全部件	403
10.6.3	许可证与鉴别码	404
10.6.4	认证过的 RPC	405
10.6.5	访问控制表 (ACL)	407
10.7	分布式文件系统	408
10.7.1	DFS 接口	409
10.7.2	服务器核心中的 DFS 部件	411
10.7.3	客户内核中的 DFS 部件	413
10.7.4	用户空间中的 DFS 部件	415
10.8	小结	416
习 题		417
第 11 章	读物列表与参考书目	419
11.1	阅读材料建议	419
11.1.1	介绍性和普通著作	419
11.1.2	分布式系统通信	420
11.1.3	分布式系统同步	420
11.1.4	分布式系统进程和处理机	421
11.1.5	分布式文件系统	422
11.1.6	分布式共享存储器	422
11.1.7	实例研究 1: Amoeba	422
11.1.8	实例研究 2: Mach	423
11.1.9	实例研究 3: Chorus	423
11.1.10	实例研究 4: DCE	423
11.2	文献目录 (按字母顺序排列)	424

第 1 章 分布式系统概述

计算机系统正在经历着一场革命。从 1945 年现代计算机时代开始到 1985 年前后，计算机是庞大而又昂贵的。即使是小型机，通常也每台价值数万美元。因此，大多数机构只有少数的几台计算机，同时，由于缺乏一种将它们连接起来的方法，所以这些计算机只能相互独立地运行。

但是，从 20 世纪 80 年代中期开始，技术上的两大进步开始改变这种状况。首先是功能更强的微处理机的开发，开始出现了 8 位的机型，随后不久 16 位，32 位，甚至 64 位的 CPU 也开始普及。其中许多机器具有较大主机（即大型机）的计算能力，但价格却只是它的几分之一。

在过去的半个世纪里，计算机技术取得了惊人的进步，这在其他工业中是前所未有的。从每台机器价格高达 1000 万美元，每秒执行一条指令，发展到目前售价 1000 美元而每秒执行 1000 万条指令，其性能价格比提高了 10^{11} 倍。如果在同一时期内汽车工业也能以这样的速度发展，那么现在一部劳斯莱斯牌汽车（Rolls Royce）将会只需要花 10 美元就可买到，而每加仑汽油就能行驶 10 亿英里（不幸的是，那时可能会有一本 200 页的手册告诉你该如何打开车门）。

第二个进步是高速计算机网络的出现。局域网 LAN 使得同一建筑内的数十甚至上百台计算机连接起来，使少量的信息能够在大约 1 毫秒左右的时间里在计算机间传送。更大量的数据则以 $10^7 \sim 10^8$ 比特/秒（bps）或更大的速率传送。广域网 WAN 使得全球范围内的数百万台计算机连接起来，传输速率从 64Kbps(每秒千位比特)到用于一些先进的实验型网络中的每秒千兆比特（gigabits）。

应用这些技术的结果，使得将大量 CPU 组成的计算系统通过高速网络连接在一起不仅成为可能，而且变得十分容易。相对于以前包括单个 CPU、存储器、外设和一些终端在内的集中式系统（又称单处理机系统 single processor system），它们通常被称为分布式系统(distributed systems)。

现在仅存在一个比较棘手的问题，那就是软件。分布式系统需要与集中式系统完全不同的软件。特别是系统所需要的操作系统只是刚刚出现。虽然分布式系统已经向前迈出了最初的几步，但仍有很长的一段路要走。对于分布式操作系统，我们对它的一些基本思想的介绍到这里已经足够了。接下来，本书将致力于研究分布式操作系统的概念、实现和几个实例。

1.1 什么是分布式系统？

分布式系统有很多不同的定义，但其中没有一个是令人满意或者能够被所有人接受的。介绍分布式系统，对它的特点的给予下面大致的描述足够了：

“一个分布式系统是一些独立的计算机的集合，但是对这个系统的用户来说，系统

就像一台计算机一样。”

这个定义有两个方面的含义：第一，从硬件角度来讲，每台计算机都是自主的；第二，从软件角度来讲，用户将整个系统看作是一台计算机。这两者都是必需的，缺一不可。在简要介绍有关硬件、软件的一些背景材料之后，我们将再回到这两点上来进行讨论。

由于给出分布式系统的一些实例可能要比进一步地深入研究定义更有帮助，下面就给出一些分布式系统的例子。第一个例子，设想一个大学或公司部门内的工作站网络。除了每个用户的个人工作站外，机房中可能还有一个共享的处理机池(pool of processor)，这些处理机并没有分配给特定的用户，而是在需要的时候进行动态分配。这样的系统可能会有一个单一的文件系统，其中所有的文件可以从所有的计算机上以相同的方式并且使用相同的路径名存取。另外，当一个用户输入一条命令时，系统能够找到一个最好的地方执行该命令。这可能是在用户自己的工作站上，可能是在别人空闲的工作站上，也可能在机房里一个未分配的处理机上。如果这个从系统整体上看以及运行起来看都像一个典型的单处理机分时系统，那么就可以称它为一个分布式系统。

第二个例子，考虑一个到处是机器人的工厂。每个机器人都有一台功能强大的计算机用于处理视觉、进行计划、通信以及其他任务。当装配线上的某个机器人发现一个它要安装的零件有缺陷时，它就要求该零件供应部门的另一个机器人给它送一个替代品。如果所有的机器人都如同连接于同一中心计算机上的外设一样工作，而且系统的程序也是以这种方式进行编制的话，那么它也是一种分布式系统。

最后一个例子是一个在世界各地有数百个分支机构的大银行。每个分支机构有一台主计算机存储当地帐目和处理本地事务。此外，每台计算机还能与其他分支机构的计算机及总部的计算机对话。如果不管顾客和帐目在哪里都能够进行交易，而且用户也不会感到当前这个系统与被替代的原先的集中式主机有何不同，那么这个系统也被认为是一个分布式系统。

1.2 目标

我们已经能够建立一个分布式系统，但这并不意味着建立一个分布式系统一定是一个好主意。就像以目前的技术水平，我们可以在一台计算机中装入四个软盘驱动器，而问题在于这样做并没有什么意义一样。本节中，我们将讨论典型分布式系统的动机和目的，分析它与传统的集中式系统相比较所具有的优点和缺点。

1.2.1 分布式系统与集中式系统比较而言的优点

系统倾向于分布式发展潮流的真正驱动力是经济。25年前，计算机权威和评论家 Herb Grosch 指出 CPU 的计算能力与它的价格的平方成正比，后来成为 Grosch 定理。也就是说如果你付出两倍的价钱，就能获得四倍的性能。这一论断与当时的大型机技术非常吻合，因而使得许多机构都尽其所能购买最大的单个大型机。

随着微处理机技术的发展，Grosch 定理不再适用了。现在人们只需花几百美元就能买到一个 CPU 芯片，这个芯片每秒钟执行的指令比 80 年代最大的大型机的处理器每秒钟所执行的指令还多。如果你愿意付出两倍的价钱，将得到同样的 CPU，但它却以更高的