

世界著名计算机教材精选

PEARSON
Prentice
Hall

分布式系统 原理与范型

(第2版)

Andrew S. Tanenbaum
Maarten Van Steen

著

辛春生 陈宗斌 等

译



**DISTRIBUTED SYSTEMS PRINCIPLES
AND PARADIGMS**

Second Edition

清华大学出版社

Simplified Chinese edition copyright © 2008 by PEARSON EDUCATION ASIA LIMITED and Tsinghua University Press.

Original English language title from Proprietor's edition of the Work.

Original English language title: Distributed Systems: Principles and Paradigms, 2E by Andrew S. Tanenbaum, Maarten Van Steen, Copyright © 2008

EISBN: 0-13-239227-5

All Rights Reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Prentice Hall.

This edition is authorized for sale only in the People's Republic of China (excluding the Special Administrative Region of Hong Kong and Macao).

本书中文简体翻译版由 Pearson Education(培生教育出版集团)授权给清华大学出版社在中国境内(不包括中国香港、澳门特别行政区)出版发行。

北京市版权局著作权合同登记号 图字 01-2006-6347 号

本书封面贴有 Pearson Education(培生教育出版集团)激光防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

分布式系统原理与范型(第2版)/(美)特尼博姆(Tanenbaum, A. S.)等著;辛春生等译.
—2版. —北京:清华大学出版社,2008.6

(世界著名计算机教材精选)

书名原文: Distributed Systems: Principles and Paradigms, 2E

ISBN 978-7-302-17279-6

I. 分… II. ①特… ②辛… III. 分布式操作系统—教材 IV. TP316.4

中国版本图书馆 CIP 数据核字(2008)第 042078 号

责任编辑:龙啟铭

责任校对:徐俊伟

责任印制:何 芊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

邮 编:100084

社 总 机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者:北京密云胶印厂

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185×260 印 张:32

字 数:773 千字

版 次:2008 年 6 月第 1 版

印 次:2008 年 6 月第 1 次印刷

印 数:1~3000

定 价:59.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。

联系电话:010-62770177 转 3103 产品编号:023234-01

译者序

自本书的第1版出版以来,计算机系统技术,尤其是分布式系统技术发生了巨大的变化。根据分布式系统技术的发展和实际教学的需要,作者对本书进行了补充和更新。本书分成了两大部分。第2~9章讨论的是分布式系统的原理、概念和技术,包括通信、进程、命名、同步化、一致性和复制、容错性以及安全性等,而分布式应用的开发方法(即范型)在第10~13章中进行了讨论。但是,与前一版不同的是,本书没有在讨论范型的章节中完整地介绍每个案例研究,而是通过一个有代表性的案例来解释原理。这种介绍方法不仅精简了素材,而且可以使得读者在阅读和学习时更愉快些。全书结构清晰,内容全面经典,系统性与先进性并茂。

本书适用对象广泛,不仅可以作为学习分布式计算机的本科生和研究生的教材,对于从事分布式计算研究和工程应用的科技人员和工程技术人员来说,本书也是一本优秀的读物。

参与本书翻译、审稿的人员有辛春生、陈宗斌、周成兴、张长富、陈河南、陈红霞、张景友、易小丽、陈婷、管学岗、王新彦、金惠敏、张海峰、徐晔、戴锋、张德福、张士华、张锁玲、杜明宗、高玉琢、王涛、申川、孙玲、李振国、高德杰、宫飞、侯经国、刘淑妮、张春林、李大成、程明、张路红、张淑芝、孙先国、刘冀得、梁永翔、张广东、郁琪琳、邵长凯、蒲书箴、潘曙光、刘瑞东、李军等。

由于分布式是一种快速发展的新技术,以及译者水平有限,书中难免有错误和疏漏之处,恳请读者不吝指正。

前 言

分布式系统已经成为了计算机科学的一个快速发展的领域。从本书的前一版出版以来,分布式系统又出现了一些新的主题,如点对点计算和传感器网络,而其他主题则变得更加成熟了,比如 Web 服务和 Web 应用等。这些发展变化要求我们对这本教材进行修订。

本版对上一版进行了重大修订。我们添加了单独的一章,以反映分布式系统组织结构所取得的进展。另一个重大的修改是,本版介绍了更多的非集中式系统内容,尤其是点对点计算。我们不仅讨论了基本技术,而且还介绍了具体的应用,如文件共享、信息传播、内容传送网络和发布/订购系统等。

除了这两个重大主题外,本书还介绍了其他一些新的主题。例如,我们介绍了传感器网络、虚拟化技术、服务器集群和网格计算等。我们还特别关注了分布式系统的自我管理,随着系统不断扩展,这是一个越来越重要的主题。

当然,我们还与时俱进地给出了一些新素材。例如,在讨论一致性与复制时,我们关注的重点是更适合现代分布式系统的一致性模型,而不是最初的模型。同样,我们还添加了有关现代分布式算法的素材,包括基于 GPS 的时钟同步化和本地化算法。

与前一版相比,本版的总页码有所减少。这主要是本版弃用了诸如分布式垃圾收集和电子支付协议等主题,并对后 4 章进行了重新组织。

与前一版一样,本书分成了两大部分。第 2~9 章讨论的是分布式系统的原理,而分布式应用的开发方法(即范型)在第 10~13 章中进行了讨论。但是,与前一版不同的是,我们没有在讨论范型的章节中完整地介绍每个案例研究,而是通过一个有代表性的案例来解释原理。例如,对象激活作为一个通信原理在第 10 章关于基于对象的分布式系统中进行了介绍。这种介绍方法使得我们不仅精简了素材,而且可以使得读者在阅读和学习时更愉快些。

当然,我们仍然是努力从实践来阐述什么是分布式系统。本书介绍各种的实际系统,如 Web-Sphere MQ、DNS、GPS、Apache、CORBA、Ice、NFS、Akamai、TIB/Rendzvous、Jini 等。这些示例阐明了理论与实践的关系,使得分布式系统成为了一个令人激动的领域。

许多人以多种方式对本书作出了贡献。我们尤其要感谢 D. Rober Adams、Arno Bakker、Coskun Bayrak、Jacques Chassin de Kergommeaux、Randy Chow、Michel Chaudron、Puneet Singh Chawla、Fabio Costa、Cong Du、Deck Epema、Kevin Fenwick、Chandana Gamage、Ali Ghodsi、Giorgio Ingargiola、Mark Jelasity、Ahmed Kamel、Gregory Kapfhammer、Jeoren Ketema、Onno Kubbe、Patricia Lago、Steve MacDonald、Michel J. Bramanian、Chintan Shah、Ruud Stegers、Paul Tymann、Craig E. Wills、Reyven Yagel 和

Dakai Zhu 阅读了部分书稿,帮助修订了前一版中的一些错误,并提出了宝贵意见。

最后,我们还要感谢我们的家庭。Suzanne 已经经历过很多次这样的情况了。她从未说过“我受够了”,尽管这个念头肯定在她脑海里出现过。谢谢你! Barbara 和 Marvin 现在对教授们为谋生所做的工作有了更好的了解,并且认识到好教材和坏教材之间的差别。现在,对我来说他们是我努力创作出更多好教材的动力所在。

目 录

第 1 章 概述	1
1.1 分布式系统的定义	1
1.2 目标	3
1.2.1 使资源可访问	3
1.2.2 透明性	4
1.2.3 开放性	6
1.2.4 可扩展性	7
1.3 分布式系统的类型	12
1.3.1 分布式计算系统	12
1.3.2 分布式信息系统	14
1.3.3 分布式普适系统	17
本章小结	21
习题	22
第 2 章 体系结构	23
2.1 体系结构的样式	23
2.2 系统体系结构	25
2.2.1 集中式体系结构	25
2.2.2 非集中式体系结构	30
2.2.3 混合体系结构	36
2.3 体系结构与中间件	38
2.3.1 中断器	38
2.3.2 自适应软件的常见方法	40
2.3.3 讨论	40
2.4 分布式系统的自我管理	41
2.4.1 反馈控制模型	42
2.4.2 示例：用 Astrolabe 监视系统	43
2.4.3 示例：Globule 中的差分复制策略	44
2.4.4 示例：Jade 的自动组件修复管理	45
本章小结	46
习题	47



第3章 进程	48
3.1 线程	48
3.1.1 线程简介	49
3.1.2 分布式系统中的线程	52
3.2 虚拟化	55
3.2.1 虚拟化在分布式系统中的作用	55
3.2.2 虚拟机体系结构	56
3.3 客户	57
3.3.1 网络连接的用户接口	58
3.3.2 客户端软件与分布透明性	61
3.4 服务器	62
3.4.1 常见的设计问题	62
3.4.2 服务器集群	65
3.4.3 管理服务器集群	69
3.5 代码迁移	72
3.5.1 代码迁移方案	72
3.5.2 迁移与本地资源	75
3.5.3 异构系统中的代码迁移	77
本章小结	79
习题	80
第4章 通信	81
4.1 基础知识	81
4.1.1 分层协议	82
4.1.2 通信类型	87
4.2 远程过程调用	89
4.2.1 基本的RPC操作	89
4.2.2 参数传递	92
4.2.3 异步RPC	95
4.2.4 示例: DCE RPC	96
4.3 面向消息的通信	101
4.3.1 面向消息的瞬时通信	101
4.3.2 面向消息的持久通信	104
4.3.3 示例: IBM WebSphere 消息队列系统	109
4.4 面向流的通信	113
4.4.1 为连续媒体提供支持	114
4.4.2 流与服务质量	115
4.4.3 流同步	117
4.5 多播通信	120
4.5.1 应用层多播	120

4.5.2 基于 gossip 的数据通信	123
本章小结	126
习题	127
第 5 章 命名系统	129
5.1 名称、标识符和地址	129
5.2 无层次命名	131
5.2.1 简单方法	131
5.2.2 基于宿主位置的方法	134
5.2.3 分布式散列表	135
5.2.4 分层方法	138
5.3 结构化命名	141
5.3.1 名称空间	141
5.3.2 名称解析	143
5.3.3 名称空间的实现	146
5.3.4 示例：域名系统	151
5.4 基于属性的命名	157
5.4.1 目录服务	157
5.4.2 分层实现：LDAP	158
5.4.3 非集中式实现	161
本章小结	164
习题	165
第 6 章 同步化	167
6.1 时钟同步	167
6.1.1 物理时钟	168
6.1.2 全球定位系统	171
6.1.3 时钟同步算法	172
6.2 逻辑时钟	176
6.2.1 Lamport 逻辑时钟	176
6.2.2 向量时钟	179
6.3 互斥	182
6.3.1 概述	182
6.3.2 集中式算法	182
6.3.3 非集中式算法	183
6.3.4 分布式算法	184
6.3.5 令牌环算法	186
6.3.6 四种算法的比较	187
6.4 结点的全局定位	188
6.5 选举算法	190



6.5.1 传统的选举算法.....	190
6.5.2 无线系统环境中的选举算法.....	192
6.5.3 大型系统中的选举算法.....	193
本章小结.....	195
习题.....	196
第7章 一致性和复制.....	198
7.1 概述	198
7.1.1 进行复制的原因.....	198
7.1.2 作为扩展技术的复制.....	199
7.2 以数据为中心的一致性模型	200
7.2.1 持续一致性.....	201
7.2.2 一致的操作顺序.....	204
7.3 以客户为中心的一致性模型	209
7.3.1 最终一致性.....	209
7.3.2 单调读.....	211
7.3.3 单调写.....	212
7.3.4 读写一致性.....	213
7.3.5 写读一致性.....	214
7.4 复制管理	214
7.4.1 副本服务器的放置.....	215
7.4.2 内容复制与放置.....	216
7.4.3 内容分发.....	219
7.5 一致性协议	222
7.5.1 持续一致性.....	222
7.5.2 基于主备份的协议.....	223
7.5.3 复制的写协议.....	225
7.5.4 高速缓存相关性协议.....	227
7.5.5 实现以客户为中心的一致性.....	228
本章小结.....	230
习题.....	231
第8章 容错性.....	233
8.1 容错性概述	233
8.1.1 基本概念.....	233
8.1.2 故障模式.....	235
8.1.3 使用冗余掩盖故障.....	237
8.2 进程恢复	238
8.2.1 设计问题.....	238
8.2.2 故障掩盖和复制.....	240

8.2.3	故障系统的协定	240
8.2.4	故障检测	243
8.3	可靠的客户-服务器通信	244
8.3.1	点对点通信	244
8.3.2	失败时的 RPC 语义	244
8.4	可靠的组通信	248
8.4.1	基本的可靠多播方法	249
8.4.2	可靠多播中的可扩展性	249
8.4.3	原子多播	252
8.5	分布式提交	257
8.5.1	两阶段提交	257
8.5.2	三阶段提交	262
8.6	恢复	263
8.6.1	概述	263
8.6.2	检查点	266
8.6.3	消息日志	268
8.6.4	面向恢复的计算	270
	本章小结	271
	习题	271
第 9 章	安全性	273
9.1	安全性概述	273
9.1.1	安全威胁、策略和机制	273
9.1.2	设计问题	277
9.1.3	加密	281
9.2	安全通道	287
9.2.1	身份认证	287
9.2.2	消息的完整性和机密性	293
9.2.3	安全组通信	295
9.2.4	示例: Kerberos	298
9.3	访问控制	299
9.3.1	访问控制中的常见问题	299
9.3.2	防火墙	302
9.3.3	安全的移动代码	304
9.3.4	拒绝服务	309
9.4	安全管理	310
9.4.1	密钥管理	310
9.4.2	安全组管理	313
9.4.3	授权管理	314
	本章小结	318



习题	319
第 10 章 基于对象的分布式系统	320
10.1 体系结构	320
10.1.1 分布式对象	320
10.1.2 示例: 企业级 Java Bean	322
10.1.3 示例: Globe 分布式共享对象	324
10.2 进程	325
10.2.1 对象服务器	325
10.2.2 示例: Ice 运行时系统	327
10.3 通信	329
10.3.1 把客户绑定到对象上	329
10.3.2 静态远程方法调用与动态远程方法调用	330
10.3.3 参数传递	331
10.3.4 示例: Java RMI	332
10.3.5 基于对象的消息传递	334
10.4 命名	337
10.4.1 CORBA 对象引用	337
10.4.2 Globe 对象引用	338
10.5 同步	340
10.6 一致性与复制	341
10.6.1 入口一致性	341
10.6.2 复制的调用	343
10.7 容错性	344
10.7.1 示例: CORBA 的容错性	345
10.7.2 示例: Java 的容错性	346
10.8 安全性	348
10.8.1 示例: Globe	348
10.8.2 远程对象的安全性	351
本章小结	352
习题	353
第 11 章 分布式文件系统	354
11.1 体系结构	354
11.1.1 客户-服务器体系结构	354
11.1.2 基于群集的分布式文件系统	358
11.1.3 对称式体系结构	360
11.2 进程	361
11.3 通信	362
11.3.1 NFS 中的 RPC	362

11.3.2	RPC2 子系统	363
11.3.3	Plan 9 中面向文件的通信	365
11.4	命名	366
11.4.1	NFS 中的命名	366
11.4.2	构造全局名称空间	370
11.5	同步	371
11.5.1	文件共享的语义	371
11.5.2	文件锁定	373
11.5.3	在 Coda 中共享文件	375
11.6	一致性和复制	376
11.6.1	客户端缓存	376
11.6.2	服务器端复制	379
11.6.3	对等文件系统中的复制	381
11.6.4	网格系统中的文件复制	382
11.7	容错性	383
11.7.1	处理 Byzantine 故障	383
11.7.2	对等系统中的高度可用性	385
11.8	安全性	386
11.8.1	NFS 中的安全性	386
11.8.2	分散式身份认证	388
11.8.3	安全的对等文件共享系统	391
	本章小结	393
	习题	394
第 12 章	基于 Web 的分布式系统	395
12.1	体系结构	395
12.1.1	传统的基于 Web 的系统	396
12.1.2	Web 服务	399
12.2	进程	402
12.2.1	客户	402
12.2.2	Apache Web 服务器	403
12.2.3	Web 服务器群集	405
12.3	通信	406
12.3.1	超文本传输协议	407
12.3.2	简单对象访问协议	411
12.4	命名	412
12.5	同步	414
12.6	一致性与复制	414
12.6.1	Web 代理缓存	415
12.6.2	Web 宿主系统的复制	417

12.6.3	Web 应用程序的复制	421
12.7	容错性	423
12.8	安全性	424
	本章小结	425
	习题	426
第 13 章	基于协作的分布式系统	428
13.1	协作模型介绍	428
13.2	体系结构	430
13.2.1	一般方法	430
13.2.2	传统的体系结构	431
13.2.3	对等体系结构	434
13.2.4	移动性和协作	436
13.3	进程	437
13.4	通信	437
13.4.1	基于内容的路由	437
13.4.2	支持复合订阅	439
13.5	命名	439
13.5.1	描述复合事件	440
13.5.2	匹配事件与订阅	441
13.6	同步	442
13.7	一致性和复制	442
13.7.1	静态方法	442
13.7.2	动态复制	445
13.8	容错性	447
13.8.1	可靠的发布-订阅通信	447
13.8.2	共享数据空间中的容错性	449
13.9	安全性	450
13.9.1	保密性	450
13.9.2	安全的共享数据空间	452
	本章小结	452
	习题	453
第 14 章	补充读物与参考文献	454
14.1	进一步阅读的建议	454
14.1.1	介绍性和综述性的著作	454
14.1.2	体系结构	455
14.1.3	进程	455
14.1.4	通信	456
14.1.5	命名	456



14.1.6	同步化.....	457
14.1.7	一致性与复制.....	457
14.1.8	容错性.....	458
14.1.9	安全性.....	458
14.1.10	面向对象的分布式系统	459
14.1.11	分布式文件系统	459
14.1.12	基于 Web 的分布式系统	460
14.1.13	基于协作的分布式系统	460
14.2	参考文献.....	461

第1章

概述

计算机系统正在经历一场革命。1945 年是现代计算机时代的开始。从那时起一直到 1985 年,计算机一直是庞大笨重而昂贵的。即便是小型机每台也要上万美元,因此,大多数机构都只有很少的几台计算机,而且由于没有把这些计算机连接起来的手段,所以只能单独使用它们。

然而,从 20 世纪 80 年代中期开始,技术领域中两方面的进步开始使这种局面有所改观。第一项进步是高性能微处理器的开发。起初这些机器是 8 位机,但是很快 16 位、32 位乃至 64 位的 CPU 纷至沓来,并得到了普及。很多使用这些 CPU 的机器的计算性能都可以与大型机相媲美,而价格却比大型机便宜得多。

近半个世纪以来,在计算机技术改进方面所取得的成果数量之多对于其他工业来说是完全无法想象的。一台计算机的价格从 1 亿美元下降到了 1000 美元,而运算速度却从每秒 1 条指令提升到每秒 1 千万条指令,性能价格比提升了大约 10^{12} 倍。如果汽车的性能价格比在同一时期以同样的速率提升,那么一辆“劳斯莱斯”牌汽车现在只值 1 美元,而且每消耗 1 加仑(1 加仑=4.5461 升)汽油可以行驶 10 亿英里(1 英里=1.6093 公里)(但是,如果果真如此,光是说明如何打开车门这一件事,可能在用户手册中就要占用长达 200 页的篇幅)。

第二项进步是高速计算机网络的发明。利用局域网技术可以将位于同一座建筑物里的数百台机器连接起来。使用这种连接方式可以在几 μs 内将少量数据从一台机器传输到另一台机器。如果要传输大量的数据,传输速率可以达到 $10\sim 1000\text{Mb/s}$ 。利用广域网技术可以连接全球范围内数百万台机器,机器间的数据传输速率从 64Kb/s 到若干 Gb/s 不等。

现在,由于有以上这些技术可供使用,因此把若干由大量计算机组成的计算系统彼此通过高速网络相连接,不但是可行的,而且是很容易实现的。这种系统一般称为计算机网络或者分布式系统,以突出其与传统的集中式系统(centralized system,也称为单处理器系统(single processor system))的差异。传统的集中式系统一般由单个计算机及其外围设备构成,也可以包含一些远程终端。

1.1 分布式系统的定义

已经有很多文献给出了分布式系统的定义,但是不同文献给出的定义彼此不尽相同,而且没有一种令人满意。考虑到本书的目标,我们只给出一个粗略的描述:

分布式系统是若干独立计算机的集合,这些计算机对于用户来说就像是单个相关系统。

这个定义包含了两方面的内容。第一个方面是关于硬件的:机器本身是独立的。第二个方面是关于软件的:对用户来说他们就像在与单个系统打交道。这两个方面共同阐明了分布式系统的本质,缺一不可。我们先介绍关于软硬件的一些背景材料,在本章稍后再回到这两个要点上来。

与继续探讨分布式系统的定义相比,把注意力集中在分布式系统的重要特性上可能更好一些。其重要特性之一是,各种计算机之间的差别以及计算机之间的通信方式的差别对用户是隐藏的。同样,用户也看不到分布式系统的内部组织结构。另一个重要特性是,用户和应用程序无论在何时何地都能够以一种一致和统一的方式与分布式系统进行交互。

分布式系统的扩展或者升级应该是相对比较容易的。这一特性是由于下述原因:分布式系统由独立的计算机组成,但同时隐藏了其中单个计算机在系统里所承担任务的细节。即使分布式系统中某些部分可能暂时发生故障,但其整体在通常情况下总是保持可用。用户和应用程序不会察觉到哪些部分正在进行替换和维修,以及加入了哪些新的部分来为更多的用户和应用程序提供服务。

为了使种类各异的计算机和网络都呈现为单个的系统,分布式系统常常通过一个“软件层”组织起来,该“软件层”在逻辑上位于由用户和应用程序组成的高层与由操作系统组成的低层之间,如图 1.1 所示。这样的分布式系统有时又称为**中间件(middleware)**。

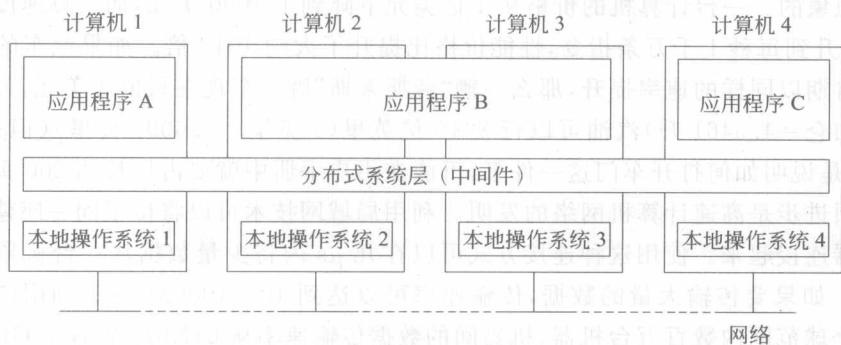


图 1.1 作为中间件组织的分布式系统。中间件层延伸到了多台机器上,且为每个应用程序提供了相同的接口

现在让我们考察一下分布式系统的几个例子。第一个例子是位于一所大学或者某个公司部门里的工作站网络。该系统除了包括每个用户自己的工作站以外,还应该包括机房内的一个处理器池。这些处理器并不是分配给特定的用户,而是根据需要进行动态调配。这样的系统可能包含一个单一的文件系统,允许所有的机器通过相同的方法并且使用相同的路径名来访问所有文件。并且,当用户输入一个命令的时候,系统将寻找执行该命令的最佳位置,可能会在用户自己的工作站上直接执行该命令,也可能在别人的一个空闲工作站上执行,还有可能由机房中某个尚未分配的处理器执行。如果系统整体外观和行为与传统的单处理器分时系统(即多用户系统)相似,那么这个系统就可以看作是一个分布式系统。

第二个例子是某个工作流信息系统,该系统支持对订单的自动处理。一般情况下,会有

来自多个不同部门的人员在不同的地点使用这样的系统。例如,销售部人员可能遍布在很大的一个区域,甚至全国。可以通过电话网络(或者蜂窝电话)连接到系统的膝上型计算机来下达订单。收到的订单由系统自动传送到计划部,接着新的内部调货订单就会送达仓储部,同时由财务部处理账单。该系统自动将订单传送到相关人员手中。用户根本看不到系统中订单处理的物理流程;对于用户来说,这些订单像是由一个集中式数据库处理的一样。

最后一个例子是万维网。它提供了一种简单、一致并且统一的分布式文档模型。要查看某个文档,用户只需激活一个引用(即链接),文档就会显示在屏幕上。理论上(但是目前在实际中并不是这样)并不需要知道该文档来自于哪个服务器,更用不着关心服务器所在的位置。要发布一个文档也很简单:只需要赋给它一个唯一的 URL 名,让该 URL 指向包含文档内容的本地文件即可。如果万维网向用户呈现的是一个庞大的集中式文档系统,也可以认为它是一个分布式系统。但是,我们现在还无法做到这一点。例如,用户明白这样一个事实:文档位于不同的地点,并由不同的服务器处理。

1.2 目 标

如果仅仅是因为有能力组建分布式系统就去组建它,这并不是明智之举。毕竟,以现在的技术,可以在一台个人计算机上安装 4 个软驱。但这样做毫无意义。在本节中,我们将讨论要使得一个分布式系统真正物有所值必须具有的 4 个关键目标,即分布式系统必须能够让用户方便地访问资源;必须隐藏资源在一个网络上分布这样一个事实;必须是开放的;必须是可扩展的。

1.2.1 使资源可访问

分布式系统的最主要目标是使用户能够方便地访问远程资源,并且以一种受控的方式与其他用户共享这些资源。资源几乎可以是任何东西,其典型例子包括打印机、计算机、存储设备、数据、文件、Web 页以及网络,但这些只是资源中的一小部分。有很多理由要求资源共享,一个显而易见的理由是降低经济成本。例如,让若干用户共享一台打印机比为每一位用户购买并维护一台打印机要划算得多。基于同样原因,共享超级计算机和高性能存储系统这样昂贵资源也是很有意义的。

用户连接到资源以后,协作和信息交换也会变得更加方便,在这方面因特网是一个最好的例子。它通过采用交换文件、邮件、文档、音频以及视频等数据的简单协议而获得了成功。因特网的连接能力造就了众多的虚拟团体,这些团体由地域上分散的人们组成,他们通过群件(groupware)协同工作。群件是支持协作编辑、远程会议等活动的软件。因特网的连接能力还使电子商务成为可能,现在通过电子商务我们不用去商店就可以买卖各种货物。

然而,在连接能力和共享功能不断加强的同时,安全性也变得愈发重要。就当前的实际情况而言,系统在通信窃听或入侵方面进行的防范是极为有限的。密码以及其他敏感信息常以明文(未加密)形式通过网络发送或者存储在服务器上,我们只能期望这些服务器是可信的。从这个意义上来说,还有许多需要改进的地方。例如,现在只需要提供信用卡号码