中国科学出版集团
新世纪书局

PEARSON

技术经典
著作大系
信息科学

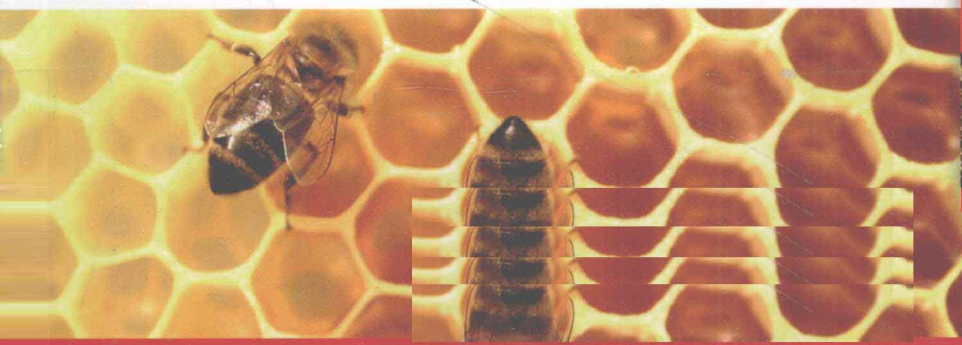
[美] Douglas C. Schmidt, Stephen D. Huston / 编著
叶 斌 / 译

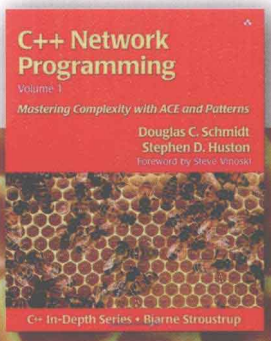
C++网络编程

卷1：运用ACE和模式消除复杂性

C++ Network Programming

Volume 1 Mastering Complexity with ACE and Patterns





世界级大师提供的专业工具，为您解决并发网络应用的开发难题

科学出版社

C++网络编程

卷1：运用ACE和模式消除复杂性

C++ Network Programming

Volume 1: Mastering Complexity with ACE and Patterns

[美] Douglas C. Schmidt, Stephen D. Huston 编著

叶斌 译

科学出版社

图字：01-2011-6670 号

内 容 简 介

本书提供了一种应用 ACE 和能够运行于多种硬件平台和操作系统的开发源框架，来开发和优化复杂分布式系统的实际解决方案，指导软件开发人员开发高效、可移植和灵活的并发式网络应用。书中通过一个贯穿全文的案例——网络日志服务，具体演示了 ACE 在并发式面向对象网络编程中的应用，说明了 ACE 所拥有的优势。ACE 软件和书中描述的所有示例应用程序都是开源的，可从 http://www.riverace.com 站点上下载。

本书对想了解和掌握如何应用 C++和面向对象设计技术，从策略和技术上进行并发网络应用设计的软件工程师、研究生和高年级本科生具有很高的指导价值。

图书在版编目（CIP）数据

C++网络编程. 第 1 卷，运用 ACE 和模式消除复杂性/（美）施密特（Schmidt, D. C.），（美）休斯顿（Huston, S. D.）编著；叶斌译. —北京：科学出版社，2011.11

ISBN 978-7-03-032799-4

Ⅰ. ①C… Ⅱ. ①施… ②休… ③叶… Ⅲ. ①C 语言—程序设计 Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字（2011）第 234774 号

责任编辑：何 武 何立兵 / 责任校对：杨慧芳

责任印刷：新世纪书局 / 封面设计：张世杰

科学出版社 出版

北京东黄城根北街 16 号

邮政编码：100717

http://www.sciencep.com

中国科学出版集团新世纪书局策划

三河市李旗庄少明装订厂印刷

中国科学出版集团新世纪书局发行 各地新华书店经销

*

2012 年 1 月 第 一 版 开本：16 开

2012 年 1 月第一次印刷 印张：19

字数：462 000

定价：59.00 元

（如有印装质量问题，我社负责调换）

前　言　Preface

写这篇序言的时候，我正在欧洲各地旅游，依靠的是欧洲出色的公共交通基础设施。作为一个美国人，我着迷并讶异于这些基础设施。无论在哪个机场着陆，我都能够很容易就搭上火车或是巴士，这些交通工具快捷、干净、可靠、准时，并且可能最重要的一点是，它们能够直接到达我的目的地。离站和到站的公告使用了多种语言，标志和方向指示都很容易理解，甚至对我这样一个不说当地母语的人来说也是如此。

我在波士顿地区工作和生活，像大多数美国人一样，我几乎是完全依赖于我的车从一个地方到另一个地方。除了偶尔会用到波士顿的地铁系统之外，我基本上都是开车到各处去，因为公共交通基础设施延伸的范围极为有限，无法把我送达目的地。由于波士顿与其他地区的数百万人都处于这同一个困境中，因此我们的公路设施现在支撑着早已超出了其能应付的交通量。我知道，如果我能够确切地了解自己生命中的多少时间因坐等交通拥堵而被白白浪费了的话，那么我会非常震惊。

在网络化计算系统和交通运输系统之间存在着一些有趣的相似之处，其中最明显的地方是，这两个系统的成功都取决于具有可伸缩性的基础设施。可伸缩的交通运输系统不仅包括显而易见的基础元素，诸如火车和铁轨、或是飞机和机场，它们还需要调度、路线、维修、票务以及监控等，所有的这些方面都必须要随物理的交通运输系统自身来做调整。同样，网络化计算不仅需要主机和网络——物理上的计算和通信基础设施——且还需要基于软件的调度、路由、分发、配置、版本控制、认证、授权以及监控等，这些方面允许网络化计算系统在需要时进行调整。

关于基础设施有一个具讽刺意味的实情：要做好是极端困难的事情，然而只有对用户越加透明，让用户感觉不到它的存在，人们才会认为它是成功的。比如，尽管瑞士的阿尔卑斯山地形崎岖，但是为数不多的一些建筑师、工程师和建设者却能运用他们的专长来提供一个高效的交通运输系统，让瑞士数百万居民得以很方便地每日使用。实际上，该系统是如此可靠和易于使用，于是你很快就会觉得它本该如此，对于你来说，这一系统已经透明化了。例如，在登上瑞士的铁路系统时，你要关注的仅是从一个站点到另一个站点，而无须关心用来把你送去那儿的运输机器。除非你是一位游客，否则你可能不会去注意一些发生的事情，比如你正在穿越一个花了数年时间设计和建造的隧道；或是正在向上爬一个很陡的斜坡，因此轨道加上了一条齿轨来帮助列车做攀爬。这一铁路系统完美地实现了它应该要做的，因此你甚至不会怎么关注它。

本书是一本关于基础软件的书籍，对于网络化计算系统来说，基础软件通常被称为中间件（middleware）。它被称为中间件是因为它相当于“沙漏的腰”，驻留在操作系统和网络的上面，但位于应用程序之下。中间件以各式各样的外观、规模和功能出现，范围从J2EE应用服务器、异步消息系统和CORBA ORB到监控小型嵌入式系统的套接字的软件。中间件必须支持日益增多的各种应用、操作系统、网络协议、编程语言和数据格式。没有中间件，要征服网络化计算系统中日益增加的多样性和异构性将会是一件繁琐、易于出错且代价昂贵的事情。

前　言 Preface

尽管中间件有着各种各样的类型，中间件解决的是各式各样的问题，但不同类型的中间件倾向于使用同样的模式和共同的抽象来克服复杂性。例如，如果你打算窥探一个可伸缩的、灵活的应用服务器、消息系统或CORBA ORB的内部，你可能会发现，它们在诸如连接管理、并发、同步、事件多路分离、事件句柄分发、错误日志以及监控等任务方面都使用了类似的技术。正如使用瑞士铁路的人数要比设计和建造它的人数多得多一样，成功的中间件的用户数也是远超设计和构建该中间件的人数的。如果要设计、构建或是使用中间件，那么你的成功取决于对这些共同模式和抽象的认识、理解和运用。

虽然许多人都明白中间件需要可扩展性和灵活性，但很少有中间件能像Doug Schmidt和Steve Huston在本书中描述的自适应通信环境（ADAPTIVE Communication Environment，ACE）那样有效地提供这些特性。ACE是一个被广泛使用的C++工具包，它使用了在各种非常成功的中间件和网络应用中常见的模式和抽象。ACE已经成为许多网络化计算系统的基础，这些系统的范围从实时的航空电子系统到为大型主机的端到端通信提供支持的CORBA ORB。

像所有优秀的中间件一样，ACE把位于其之下的多样化和异构环境的复杂性隐藏了起来。然而，让ACE从大多数其他的基础中间件中脱颖而出的则是这样一个事实：尽管它支持应用在任何一种情形中所需的最大灵活性，但不会因此而降低系统的性能或是可扩展性。我自己作为一个有年头的中间件架构师，非常清楚地知道，在同一个软件包中既要能获得性又要实现灵活性是很难的。

不过，从某种意义上说，ACE在灵活性和性能方面的表现不会让我感到吃惊，基于我与Doug的长期交往，因此我很清楚，他是这一领域的先驱者。当前广泛存在的各类可扩展的、高性能的、灵活的中间件都很明显地带有他的印记，显示了他的影响力。他的工作伙伴Steve，是一名很有天赋的C++开发者和作者，他这些年的工作给ACE带来了许多改进。他们合作所取得的这一成果对于每个涉及设计、构建甚至是使用中间件的人来说，都是“必读”的经典。万维网和互连嵌入式系统的日益普及，意味着网络化计算系统的数量、规模和重要性都将持续增长。只要理解了Doug和Steve在本书中描述的这些关键的模式、技术、类和经验教训，我们就有希望能够通过提供中间件基础设施来使这一过程变得完全透明、高效且可靠。

Steve Vinoski
Platform Technologies首席架构师兼副总裁
IONA Technologies
2001年9月

关于本书 About This Book

在过去的10年中，并发式面向对象网络编程已经作为一种高效的应用软件开发范式（paradigm）出现，这些软件的协作对象可以是：

- 在一个进程或者计算机内部协作。
- 分布在一组通过网络连接的计算机上，诸如嵌入式互连系统、局域网络（LAN）、企业内部网或是互联网等。

当对象分布在不同地方时，构成这些对象的各种实体彼此之间必须要进行有效的通信和协作。而且，在应用随着其生存期发生改变时，它们也必须能够继续进行这样的工作。对象的布局、可用的网络基础设施以及平台并发机制的选择都预留了某种程度的自由选择空间，这一自由空间是功能强大的，也是充满挑战的。

在设计得当时，并发式面向对象网络编程能力可以给你的应用增加极大的灵活性。例如，按照项目的需求和可用的资源，你可能会用到：

- 实时、嵌入式或手持式系统。
- 个人计算机或笔记本电脑。
- 按各种规模分类的UNIX或Linux系统。
- “大铁块”状的主机甚至是超级计算机。

不过，在多种操作系统（OS）平台上开发和移植网络应用的时候，你可能会遇到一些错综复杂的挑战。这些复杂性会以这样的形式体现出来：网络协议不兼容，或是组件库在不同的硬件和软件平台（platform）上有着不同的API和语义，以及由于本地（native）的OS进程间通信（IPC）和并发机制本身的局限性而引入的一些偶发复杂性（accidental complexity）。为了缓解这些问题，自适应通信环境（ADAPTIVE Communication Environment，ACE）提供了一个面向对象的工具包，该工具包以可移植的方式运行在几十种硬件和OS平台上，其中包括Win32和UNIX的大多数版本，以及众多的实时和嵌入式操作系统。

可能有些人会希望你相信，一些事实上的或是官方承认的OS标准，比如POSIX、UNIX98或Win32等，它们是所有的编程者都需要用来让他们的应用免遭移植性挑战的操作系统。不幸的是，有这样一句名言：“标准的好处就是备选项很多”[Tan96]，相对于10年前来说，这句话更适用于今天。目前存在着几十种不同的应用于商业、学术和政府项目的OS平台，而且其数量还在随着每种新版本和变体的出现而不断增长。

在过去的20年中，我们已经开发了许多的多平台并发式网络系统。因此，我们可以向你确保一件事情，那就是在不同的时期，OS提供商常常会选择实现不同的标准，而且，标准也会发生变化和发展。你可能要在多个平台上工作，这些平台在不同的时期以不同的方式实现了不同的标准。因此，直接在OS的API上编程会带来以下两个问题。

- 容易出错。因为本地的OS API是用C编写的，它们往往缺乏类型安全的、可移植的、可重入的和可扩展的系统功能接口和函数库。例如，被广泛使用的Socket API

（在第2章中讨论）的通信端点是经由弱类型的整型或指针I/O句柄（handle）来标识的，这一做法增加了在运行时发生不易察觉的编程性错误的可能性。

- 助长了不合格设计技术的使用。因为许多使用OS API来编写的网络应用都是基于算法设计而不是基于面向对象设计的。算法设计根据具体的功能需求来分解应用的结构，而这些功能需求是不稳定的，且可能会随时间改变。因此这种设计范式会产生无扩展能力的软件架构，这样的架构不能通过定制来快速地满足不断变化的应用需求[Boo94]。

在这个经济动荡、管制宽松和全球性竞争剧烈的年代，使用本地的OS API和算法设计技术完全从头开始开发应用的做法已经变得代价昂贵而且耗时。

如果已经从事了多年网络软件系统的开发，你可能已经学会了把这些问题视作生活中的一个事实而给予接受。然而，还存在着一个更好的处理这些问题的办法。在本书中，我们展示了C++和ACE如何通过提供面向对象的能力来让你避免许多的困境和缺陷，但依然会利用到“标准”——甚至是某些平台特有的功能特性——在任何可能的时候。面向对象的设计展示了在随时间推移的过程中它比算法设计更强大的稳定性，这使它成为了在开发多种类型的网络应用时的首选手段。

没有什么可奇怪的，获得所有的这些灵活性需要付出一些代价：你可能需要学习一些新的概念、方法、模式、工具以及开发技术等。根据你的技术背景，这一学习曲线可能很平坦，也可能在最初时看起来很陡峭。不过，面向对象范式能够给你提供一套成熟的技术，以缓解网络应用开发带来的诸多挑战。本书给出了一系列具体的例子，以说明用来开发和应用ACE工具包中的类的面向对象技术。你可以使用同样的技术和ACE类来简化自己的应用。

目标读者

本书面向那些“动手实践型”的开发者和大学的高年级学生，他们有兴趣深入了解使用C++和面向对象设计进行并发式网络编程的策略和方法。我们描述的是一些关键的设计空间（design dimension）、模式和原则，这些方面是快速轻松地开发灵活高效的并发式网络应用所需要的。我们的数量众多的C++代码示例强化了这些设计概念，并具体地说明了如何尽快掌握ACE中的核心类。我们还会把你带到“幕后”，以便让你理解ACE工具包中的IPC和并发机制的设计方式及缘由。这些资料有助于提高你的设计技能，并能够帮助你在自己的面向对象网络应用中更加有效地运用C++和设计模式（pattern）。

本书并不是一本关于面向对象开发、模式、UML、系统编程或是网络构建的综合教程。因此我们假定这本书的读者对以下主题都有一定程度的熟悉。

- 面向对象的设计和编程技术，如框架[Joh97, FJS99b, FJS99a]、模式[GHJV95, BMR+96, SSRB00]、模块化[Mey97]、信息隐藏（information hiding）[Par72]和建模[Boo94]。
- 面向对象的表示法和过程，比如说统一建模语言（Unified Modeling Language，UML）[RJB98]、极限编程[Bec00]和Rational统一过程（Rational Unified Process，RUP）[JBR99]。

- 基本的C++语言特性，比如说类、继承（inheritance）、动态绑定（dynamic binding）和参数化类型（parameterized type）[Bja00]。
- 核心系统的编程机制，比如说事件多路分离、进程和线程管理、虚拟内存，以及UNIX[Ste98, Ste99, Ste92, Lew95, KSS96, But97]和Win32[Ric97, Sol98, JO99]平台上常用的IPC机制和API。
- 网络构建方面的术语和概念，比如说TCP/IP[Ste93]、远程操作调用[Obj01]和客户/服务器架构[CS92]。

如果你想对某些主题做更多的了解，我们鼓励你使用包罗万象的参考书目来查找相关主题的信息来源。

本书也并非是一本ACE编程者的参考手册。也就是说，我们并不会对ACE中的每个类的每个方法都做出解释。欲详细了解这一层面的信息，请参阅位于http://www.riverace.com/docs/之上的使用Doxygen[Dim01]生成的详尽的ACE在线文档。相反，本书重点关注的是：

- 关键的概念、模式和C++特性，这些方面决定了成功的面向对象网络应用和中间件设计。
- 最常用的ACE TCP/IP和并发包装器外观（wrapper façade）类的背后动机和基本用法。

结构和内容

本书描述了C++和中间件如何帮助应对与开发网络应用相关的关键性挑战。我们回顾了主流OS平台上可用的核心的本地OS机制，并说明如何在ACE中运用C++和模式来把这些机制封装在类库的包装器外观中，这些包装器外观可改善应用的可移植性和健壮性。本书主要的应用案例是一个网络日志服务，该服务通过TCP/IP把日志记录从客户端应用程序传输到日志服务器上。我们使用这一服务作为一个贯穿全书的运行例子来说明以下问题。

- 具体展示C++和ACE如何帮助实现高效、可预测和可伸缩的网络应用。
- 说明关键设计和实现的考虑和解决方案，在开发你自己的并发式面向对象网络应用的时候，这些方面的问题都会显现出来。

本书由如下的11章内容组成。

- 导言——第0章，对C++网络编程作了介绍。这部分内容开始先勾勒出问题空间，并给出当应用延伸到单个进程的单个线程之外时会面临的挑战。接着我们引入了中间件各层的一个分类，并描述如何运用宿主基础中间件（host infrastructure middleware）和ACE工具包来应对常见的网络编程方面的挑战。
- 第1部分——第1～4章，概述了通信设计的可选方法，并描述了在ACE中用来对OS IPC机制进行有效编程的面向对象技术。由此产生的类构成了本书的运行例子：网络日志服务的第一个版本的基础。

关于本书 About This Book

- 第2部分——第5～10章，概述了并发设计的可选方法，并描述了在ACE中用来对OS并发机制进行有效编程的面向对象技术。

贯穿整个第1部分和第2部分，我们给出了网络日志服务的一系列成熟度不断提升的实现，以此来说明如何在实践中运用ACE的IPC和并发包装器外观。

附录A总结了用以支撑ACE IPC和并发包装器外观的类设计和实现的原则。附录B解释了ACE的起步和过去10年来开放源码的演变情况，并简述了其未来的方向。本书最后还提供了一个技术术语的词汇表、一个涉猎广泛的可供进一步研究用的参考文献列表。

相关资料

本书致力于使用特定的C++特性、模式和ACE来解决复杂性。该系列的第二卷《C++网络编程 卷2：基于ACE和框架的系统化复用》延伸了本书的范围，把ACE提供的面向对象网络编程框架也包含了进来。这些框架具体化了本书中描述的ACE包装器外观类的常见使用模式，以支持更广泛、更具扩展性的系统级复用。本书涵盖的ACE包装器外观类和《C++网络编程 卷2：基于ACE和框架的系统化复用》涵盖的ACE框架类的一个不同点在于，ACE包装器外观类仅有少量的虚方法，而ACE框架类则几乎都是虚方法。

本书基于ACE的5.2版本，该版本于2001年10月发布。ACE软件和书中描述的所有示例应用程序都是开源的，可从http://www.riverace.com站点上下载。这些站点还包含了ACE其他方面的丰富资料，诸如教程、技术文章以及本书并未涵盖的其他ACE IPC和同步机制包装器外观的概览等。我们鼓励你获取ACE的一个副本，这样你就可以完整、详细地去了解和领会实际的ACE类和框架，并可在阅读本书的过程中穿插运行相应的代码例子。ACE的预编译版本也可以以象征性的价格从http://www.riverace.com上购得。

若要了解更多关于ACE的信息，或是提交在本书中发现的错误的报告，我们建议你订阅ACE的邮件列表，地址是ace-users@cs.wust1.edu.。你可以通过发送电子邮件给ace-users-request@cs.wust1.edu向Majordomo列表服务器订阅，在电子邮件的正文中包括如下命令（主题行会被忽略）。

```
subscribe ace-users [邮箱地址@域]
```

只有在你的邮件消息的From（发件人）地址并不是你希望用来订阅的地址时，你才需要提供“邮箱地址@域”这部分内容。

提交到ACE邮件列表中的帖子也会转发到USENET的新闻组comp.soft-sys.ace上。已归档的ACE邮件列表帖文可从http://groups.yahoo.com/group/ace-users上获取。

致谢

审阅方面的优胜者荣誉要归属于Christopher Allen、Tomer Amiaz、Alain Decamps、Eric Eide、Don Hinton、Alexander Holler、Susan Liebeskind、Dennis Mancl、Craig Perras、Patrick Rabau、Eamonn Saunders和Johnny Willemsen，他们校读了全部书稿并提供了能极大改善其形式和内容的广泛建议。

其他许多来自世界各地的ACE用户就本书的手稿给出了自己的反馈意见，其中包括Mark Appel、Shahzad Aslam-Mir、Kevin Bailey、Barry Benowitz、Fang Chow、Emmanuel Croze、Yasir Faiz、Gillmer Derge、Iain Hanson、Brad Hoskins、Bob Huston、Christopher Kohlhoff、Serge Kolgan、Daire Lynch、Andy Marchewka、Jeff McNiel、Simon McQueen、Phil Mesnier、Arturo Montes、Vince Mounts、Aaron Nielsen、Jeff Parsons、Pim Philipse、Yaron Pinto、Stephane Pion、Nick Pratt、Paul Rubel、Val Salamakha、Shourya Sarcar、Chris Smith、Leo Stutzmann、Tommy Svensson、Alain Totouom、Roger Tragin、Bruce Trask、Chris Uzdavinis和Reuven Yagel。

我们非常感激华盛顿大学圣路易斯分校（Washington University, St. Louis）和加州大学欧文分校（University of California, Irvine）的DOC工作组过去和现在的所有成员，以及Object Computing公司和Riverace公司的团队成员，他们开发、改进和优化了许多本书中陈述的ACE功能。这些成员包括Everett Anderson、Alex Arulanthu、Shawn Atkins、John Aughey、Luther Baker、Darrell Brunsch、Don Busch、Chris Cleeland、Angelo Corsaro、Chad Elliot、Sergio Flores-Gaitan、Chris Gill、Pradeep Gore、Andy Gokhale、Priyanka Gontla、Myrna Harbibson、Tim Harrison、Shawn Hannan、John Heitmann、Joe Hoffert、James Hu、Frank Hunleth、Prashant Jain、Vishal Kachroo、Ray Klefstad、Kitty Krishnakumar、Yamuna Krishnamurthy、Michael Kircher、Fred Kuhns、David Levine、Chanaka Liyanaarachchi、Michael Moran、Ebrahim Moshiri、Sumedh Mungee、Bala Natarajan、Ossama Othman、Jeff Parsons、Kirthika Parameswaran、Krish Pathayapura、Irfan Pyarali、Sumita Rao、Carlos O'Ryan、Rich Siebel、Malcolm Spence、Marina Spivak、Naga Surendran、Steve Totten、Bruce Trask、Nanbor Wang和Seth Widoff。

我们还想感谢来自超过50个国家的数以千计的C++开发者。在过去10年中，他们给ACE贡献良多。ACE的卓越和成功是对许多天才开发者和具有前瞻性的公司的技能和慷慨行为的证明，他们高瞻远瞩，把自己的工作成果贡献给了ACE的开源代码库。没有他们的支持、不断的反馈和鼓励，我们绝不可能写出这本书。作为对ACE开源社区的努力成果的感谢和认可，我们制作了一份所有捐献者的名单，该名单可通过http://ace.ece.uci.edu/ACE-members.html获取。

我们还非常感激支持我们研究ACE工具包的模式和开发的同事和赞助商，特别是Ron Akers（Motorola）、Steve Bachinsky（SAIC）、John Bay（DARPA）、Detlef Becker（Siemens）、Dave Busigo（DARPA）、John Buttitto（Sun）、Becky Callison（Boeing）、Wei Chiang（Nokia）、Joe Cross（Lockheed Martin）、Lou DiPalma（Raytheon）、Bryan Doerr（Boeing）、Karlheinz Dorn（Siemens）、Matt Emerson（Escient Convergence Group, Inc.）、Sylvester Fernandez（Lockheed Martin）、Nikki Ford（DARPA）、Andreas Geisler（Siemens）、Helen Gill（NSF）、Bob Groschadl（Pivotech Systems, Inc.）、Jody Hagins（ATD）、Andy Harvey（Cisco）、Sue Kelly（Sandia National Labs）、Gary Koob（DARPA）、Petri Koskelainen（Nokia Inc）、Sean Landis（Motorola）、Patrick Lardieri（Lockheed Martin）、Doug Lea（SUNY Oswego）、Hikyu Lee（SoftLinx）、Joe Loyall（BBN）、Mike Masters（NSWC）、

关于本书 About This Book

Ed Mays（U.S. Marine Corps）、John Mellby（Raytheon）、Jeanette Milos（DARPA）、Stan Moyer（Telcordia）、Russ Noseworthy（Object Sciences）、Dieter Quehl（Siemens）、Vijay Raghavan（Vanderbilt U.）、Lucie Robillard（U.S. Air Force）、Craig Rodrigues（BBN）、Rick Schantz（BBN）、Steve Shaffer（Kodak）、Tom Shields（Raytheon）、Dave Sharp（Boeing）、Naval Sodha（Ericsson）、Paul Stephenson（Ericsson）、Tatsuya Suda（UCI）、Umar Syyid（Hughes）、Janos Sztipanovits（Vanderbilt U.）、Gautam Thaker（Lockheed Martin）、Lothar Werzinger（Krones）和Don Winter（Boeing）等。

我们特别要感谢我们的原稿编辑Susan Cooper提升了我们这些资料的书写水准。此外，我们非常感激我们的编辑Debbie Lafferty、我们的制作统筹Elizabeth Ryan、丛书编辑兼C++发明者Bjarne Stroustrup，以及Addison-Wesley的其他每一个人，是他们使本书能够出版，我们要对他们的鼓励和耐心表示感谢。

最后，我们还想向已故的W. Richard Stevens——网络编程著作第一人表达我们的谢意和感激之情。他的书给网络编程的艺术和科学带来了前所未有的清晰度。我们将尽力站在他的肩膀上，把Richard著作中的理念向面向对象设计和C++编程领域拓展延伸，让其发挥更大的作用。

Steve的致谢

我要感谢上帝，让我体会到计算机和网络带来的乐趣，我希望他也快乐。致Jane，与我生活了20年的妻子，感谢你爱我，每天都鼓励我。没有你的支持，我是不可能完成本书的——你就是神给我的祝福。感谢已故的David N. Drummond，感谢你给一个没有学位的孩子一个机会。还要感谢Doug Schmidt，是一位学者也是一位绅士，他的洞察力、积极性和创造力每天都深深地影响着我，激励着我。

Doug的致谢

花费了超过10年的时间来写这本书，因此看到它最终能够付印，我无比激动（也是一个解脱）！就这一点来说，我非常感谢Steve Huston、Debbie Lafferty和Bjarne Stroustrup的巨大帮助和耐心，最终得以看到这一项目的成果。我还要感谢我的妻子Sonja，感谢她的爱和她在我撰写本书期间对我的支持——现在书已经完成了，我们可以把更多的时间放在交谊舞上了。最后，我要感谢来自威廉与玛丽学院（College of William and Mary）、华盛顿大学圣路易斯分校和加州大学欧文分校、DARPA和Siemens的许多朋友和同事，以及感谢全球成千上万的ACE和TAO的开发者和使用者——他们极大地丰富了我过去20年来的精神生活和人际交往。

目 录

目 录

目 录

第0章

设计面临的挑战、中间件解决方案和ACE

本章提要

本章描述了从独立应用架构（stand-along application architecture）过渡到网络应用架构（networked application architecture）时所发生的范式转变。这种转变带来的挑战落于两种范畴中：面向软件架构和设计的问题空间，以及与被用来实现网络应用的软件工具和技术相关的解答空间。本章先对影响前一范畴的设计问题进行领域分析，并提出了由后一范畴引出及致力于应对该挑战的中间件；然后介绍了 ACE 工具包和贯穿本书用来说明解决方案的网络应用范例。

0.1 网络应用面临的挑战

大多数软件开发者对独立应用架构都很熟悉，在这种架构中，单台计算机包含了所有与图形用户界面（Graphical User Interface，GUI）、应用服务处理和持久数据来源相关的软件组件。例如，图 0.1 说明的独立应用架构通过与所有的外围设备直接相连，在

单台计算机上整合了GUI、服务处理和持久数据源。独立应用的控制流程完全驻留在执行开始时其所位于的计算机上。

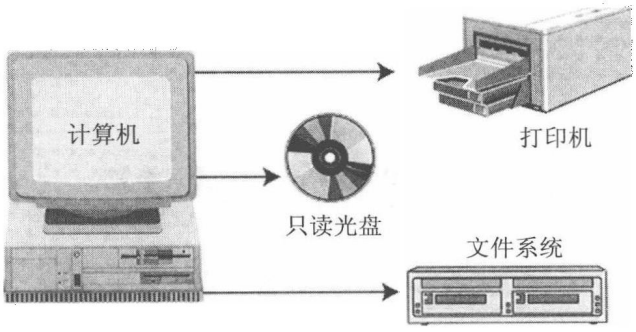


图0.1 一个独立应用的架构

与此相反，网络应用的架构把应用系统划分成多个服务（service），这些服务可被多个应用共享和重用。为了最大限度地提高效率和有效利用性，服务分布在多个通过网络连接的计算设备中，如图0.2所示。这样的环境提供给客户端（client）的网络服务通常包括分布式名称管理（distributed naming）、网络文件系统、路由表管理、日志、打印、电子邮件、远程登录、文件传输、基于Web的电子商务、付款处理、客户关系管理、服务台系统、MP3交换、流媒体、实时消息和社区聊天室等。

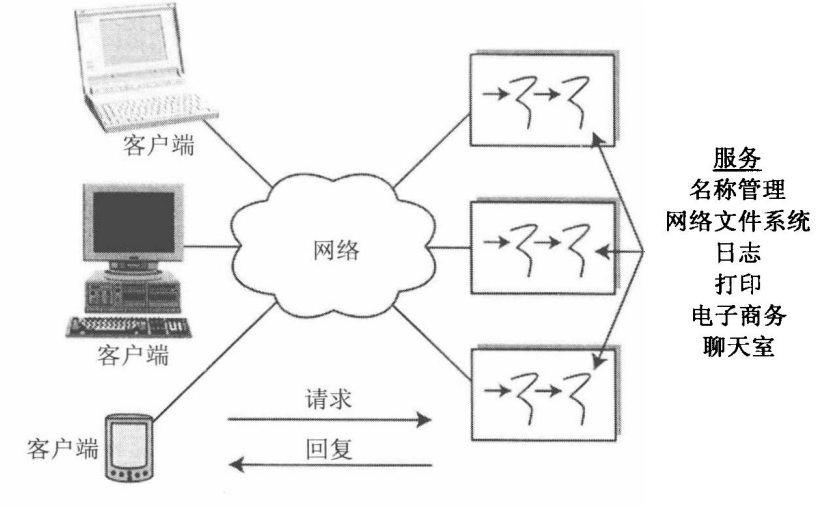


图0.2 一个常见的网络应用环境