

Programming WCF Services

精通WCF和Azure AppFabric服务总线

第三版
更新至 .NET 4.0

WCF

服务编程



Juval Löwy 著
Clemens Vasters 序
徐雷 徐扬 译

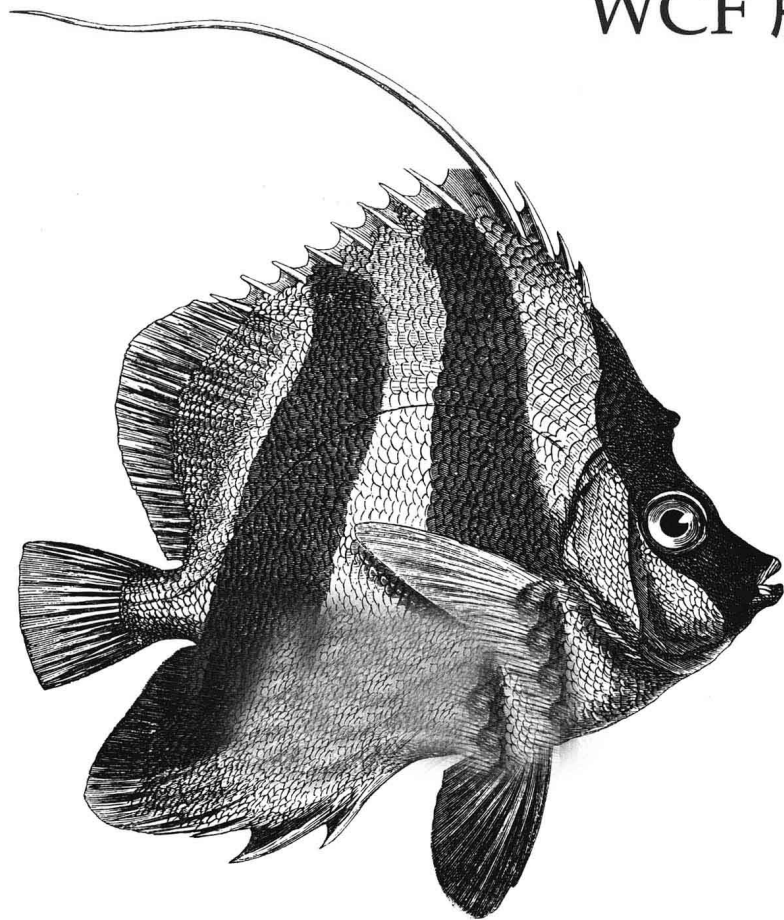
O'REILLY®



华中科技大学出版社
<http://www.hustp.com>

第三版

WCF 服务编程



Juval Löwy 著

emens Vasters 序

徐雷 徐扬 译

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

O'Reilly Media, Inc. 授权华中科技大学出版社出版

华中科技大学出版社

中国 • 武汉

本书涵盖了.NET Framework 4.0的许多新特性,以及 Azure AppFabric 服务总线,同时,还包括一些最新的 WCF 想法和技巧。本书通过告诉你“为什么”及“如何”使用 WCF 来帮助你精通 WCF 编程、成长为一名出色的软件工程师。通过本书,你可以:了解 WCF 架构和基础构件,包含可靠性和传输会话等关键概念;使用服务托管、实例和并发管理、事务、无连接队列调用、安全和服务等内置特性;精通 Windows Azure AppFabric 服务总线、新的云计算里最具革命性的部分;通过使用相关的设计原则、提示和 ServiceModelEx 框架里的最佳实践,提高 WCF 服务开发的效率和质量;挖掘服务设计背后的原理,揭秘 WCF 开发的精髓。

978-0-596-80548-7 Programming WCF Services, Third Edition © 2010 by O'Reilly Media, Inc. Simplified Chinese edition, jointly published by O'Reilly Media, Inc. and Huazhong University of Science and Technology Press, 2011. Authorized translation of the English edition, 2010 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

本书中文简体版专有版权由 O'Reilly Media, Inc. 授予华中科技大学出版社,未经许可,不得以任何方式复制或抄袭本书的任何部分。

湖北省版权局著作权合同登记图字:17-2011-102 号

图书在版编目(CIP)数据

WCF 服务编程(第三版)/Juval Löwy 著;徐雷 徐扬 译. —武汉:华中科技大学出版社,2011.6
ISBN 978-7-5609-7083-7

I. W… II. ①J… ②徐… III. 网络服务器-程序设计 IV. TP368.5

中国版本图书馆 CIP 数据核字(2011)第 090812 号

WCF 服务编程(第三版)

Juval Löwy 著
徐雷 徐扬 译

策划编辑:谢燕群

封面设计:张健

责任监印:张正林

出版发行:华中科技大学出版社(中国·武汉)

武昌喻家山 邮编:430074

电话:(027)87557487

录排:华中科技大学惠友文印中心

印刷:湖北新华印务有限公司

开本:787mm×980mm 1/16

印张:51

字数:1270 千字

版次:2011 年 6 月第 3 版第 1 次印刷

定价:118.00 元

责任编辑:江津 陈元玉

责任校对:朱玢



华中出版

本书若有印装质量问题,向出版社营销中心调换
全国免费服务热线:400-6679-118 竭诚为您服务
版权所有 侵权必究

O'Reilly Media, Inc.介绍

O'Reilly Media通过图书、在线服务、杂志、调查研究和会议等方式传播创新者的知识。自1978年开始O'Reilly一直都是发展前沿的见证者和推动者。超级极客正在开创未来，我们关注着真正重要的技术趋势，通过放大那些“微弱的信号”来刺激社会对新科技的采用。作为技术社区中活跃的参与者，O'Reilly的发展充满着对创新的倡导、创造和发扬光大。

作为出版商O'Reilly为软件开发人员带来革命性的“动物书”，创造了第一个商业网站（GNN），组织开放源代码峰会以至于开源软件运动以此命名，通过创立Make杂志成为DIY革命的主要先锋，公司一如既往地用各种方式和渠道连接人们和他们所需要的信息。O'Reilly的会议和峰会聚集了超级极客和高瞻远瞩的商业领袖，共同描绘将开创新产业的革命性思想。作为技术人士获取信息的选择O'Reilly现在还将先锋专家的知识传递给普通计算机用户。无论是通过印刷书籍、在线服务或者面授课程，每一项O'Reilly的产品都反映了公司不可动摇的信念——信息是激发创新的力量。

业界评论

“O'Reilly Radar博客有口皆碑。”

——Wired

“O'Reilly凭借一系列（真希望当初我也想到了）非凡想法建立了数百万美元的业务。”

——Business 2.0

“O'Reilly Conference是聚集关键思想领袖的绝对典范。”

——CRN

“一本O'Reilly的书就代表一个有用、有前途、需要学习的主题。”

——Irish Times

“Tim是位少有的商人，他不光放眼于最长远、最广阔的视野并且切实地按照Yogi Berra的建议去做了：“如果你在路上遇到岔路口，走小路（岔路）。”回顾过去Tim似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——Linux Journal

目录

Table of Contents

序言	I
前言	III
第 1 章 WCF 基础.....	1
什么是 WCF	1
服务	2
服务的执行边界	3
WCF 与位置透明度	4
地址	4
TCP 地址.....	5
HTTP 地址.....	5
IPC 地址	6
MSMQ 地址	6
服务总线地址.....	6
契约	6
服务契约	7
托管	11
IIS 5/6 托管	11
自托管	12
WAS 托管	17
IIS/WAS 里的自定义托管.....	18
Windows Server AppFabric.....	19
选择宿主	20
绑定	21
常用绑定	22
选择绑定	24
其他绑定	24
使用绑定	26
终结点	26
以管理方式配置终结点	27
以编程方式配置终结点	31

默认终结点	32
元数据交换	35
基于 HTTP-GET 的元数据	35
元数据交换终结点	38
元数据浏览器	44
行为配置的更多内容	46
客户端编程	47
生成代理	48
以管理方式配置客户端	52
以编程方式配置客户端	57
WCF 提供的测试客户端	58
编程方式配置 VS 管理方式配置	60
WCF 体系结构	61
宿主体系结构	62
使用通道	63
InProcFactory 类	64
传输会话层	68
传输会话与绑定	68
传输会话终止	68
可靠性	69
绑定、可靠性和有序消息	70
配置可靠性	71
必备有序传递	72
第 2 章 服务契约	75
操作重载	75
契约继承	77
客户端契约层级	78
服务契约分解与设计	82
契约分解	82
分解准则	84
契约查询	85
以编程方式处理元数据	86
MetadataHelper 类	88
第 3 章 数据契约	93
序列化	93
.NET 序列化	94
序列化的数据契约	98
数据契约特性	99

导入数据契约.....	102
数据契约与 Serializable 特性.....	104
推断数据契约.....	105
组合数据契约.....	107
数据契约事件.....	108
共享数据契约.....	110
数据契约层级.....	111
已知类型.....	112
服务已知类型.....	114
多个已知类型.....	115
配置已知类型.....	116
数据契约解析器.....	117
Objects 与接口.....	127
数据契约等效性.....	129
序列化顺序.....	130
版本控制.....	132
新增成员.....	133
缺失成员.....	133
版本控制往返.....	137
枚举.....	139
委托与数据契约.....	141
泛型.....	142
集合.....	145
具体集合类型.....	146
自定义集合.....	148
CollectionDataContract 特性.....	148
引用集合.....	150
字典集合.....	150
第 4 章 实例管理	153
行为.....	153
单调服务.....	154
单调服务的优势.....	155
配置单调服务.....	156
单调服务与传输会话.....	157
设计单调服务.....	157
选择单调服务.....	160
会话服务.....	160
配置私有会话.....	160
会话与可靠性.....	165

会话 ID	166
会话终止	167
单例服务	168
初始化单例服务	169
选择单例服务	171
操作分界	172
实例停止	174
配置为 ReleaseInstanceMode.None	176
配置为 ReleaseInstanceMode.BeforeCall	176
配置为 ReleaseInstanceMode.AfterCall	177
配置为 ReleaseInstanceMode.BeforeAndAfterCall	177
显式停止	178
使用实例停止	179
持久化服务	179
持久化服务与实例管理模式	180
实例 ID 与持久化存储区	180
显式实例 ID	181
消息头里的实例 ID	183
支持实例 ID 的上下文绑定	185
自动持久化行为	191
限流	196
配置限流	199
第 5 章 操作	205
请求/应答操作	205
单向操作	205
配置单向操作	206
单向操作与可靠性	207
单向操作与会话服务	207
单向操作和异常	208
回调操作	209
回调契约	210
客户端回调设置	211
服务端的回调调用	214
回调连接管理	217
双向代理与类型安全	219
回调契约层级	224
事件	225
流操作	229
I/O 流	229

流操作与绑定.....	230
流操作与传输.....	231
第 6 章 错误.....	233
错误隔离和解耦.....	233
错误屏蔽.....	233
通道故障.....	234
错误传播.....	238
错误契约.....	239
错误调式.....	243
错误与回调.....	249
错误处理扩展.....	252
提供错误.....	252
处理错误.....	256
安装错误处理扩展.....	258
宿主与错误扩展.....	261
回调与错误扩展.....	264
第 7 章 事务.....	267
恢复的挑战.....	267
事务.....	268
事务资源.....	268
事务属性.....	269
事务管理.....	270
资源管理器.....	273
事务传播.....	273
事务流与绑定.....	273
事务流与操作契约.....	274
单调.....	276
事务协议与管理器.....	277
协议与绑定.....	277
事务管理器.....	278
事务管理器提升.....	280
Transaction 类.....	281
环境事务.....	282
本地事务 VS 分布式事务.....	282
服务事务编程.....	283
设置环境事务.....	284
事务传播模式.....	285
投票与提交.....	292

事务隔离	295
事务超时	297
明确使用事务编程	298
TransactionScope 类	299
事务流管理	300
非服务客户端	307
服务状态管理	308
事务边界	309
实例管理与事务	309
单调事务服务	310
会话事务服务	313
事务持久化服务	326
事务行为	328
事务单例服务	333
实例模式与事务	336
回调	337
回调事务模式	338
回调投票	340
使用事务型回调	340
第 8 章 并发管理	343
实例管理与并发	343
服务并发模式	344
ConcurrencyMode.Single	344
同步访问与事务	345
ConcurrencyMode.Multiple	345
ConcurrencyMode.Reentrant	348
实例与并发访问	350
单调服务	350
会话与单例服务	351
资源与服务	352
访问死锁	352
避免死锁	353
资源同步上下文	354
.NET 同步上下文	354
UI 同步上下文	357
服务同步上下文	362
托管在 UI 线程上	363
作为服务	368
UI 线程与并发管理	371

自定义服务同步上下文	372
线程池同步化器	373
线程关联性	377
优先级处理	380
回调与客户端安全	383
使用 ConcurrencyMode.Single 回调	384
使用 ConcurrencyMode.Multiple 回调	385
使用 ConcurrencyMode.Reentrant 回调	385
回调与同步上下文	385
回调与 UI 同步上下文	386
回调自定义上下文	389
异步调用	393
异步机制的需求	393
基于代理的异步调用	394
异步调用	395
轮询或等待完成	398
完成回调	400
单向异步调用	404
异步错误调用	408
异步调用与事务	408
同步和异步调用	409
第 9 章 队列服务	411
无连接的服务与客户端	411
队列调用	412
队列调用架构	412
队列契约	413
配置与安装	414
事务	420
传递与回放	420
事务配置	422
非事务性队列	425
实例管理	425
单调队列服务	426
会话队列服务	427
单例服务	431
并发管理	432
限流	432
传送失败	433
死信队列	434

生存时间	434
配置死信队列	435
处理死信队列	436
回放失败	440
有害消息	441
MSMQ 4.0 里的有害消息处理	441
MSMQ 3.0 里的有害消息处理	446
队列调用与连接调用	446
要求队列化	447
应答服务	449
设计应答服务契约	450
客户端编程	453
队列服务端编程	455
应答服务端编程	457
事务	458
HTTP 桥	461
设计桥	461
事务配置	462
服务端配置	463
客户端配置	464
第 10 章 安全	467
验证	467
授权	468
传输安全	468
传输安全模式	469
传输安全模式配置	470
Transport 安全和凭据	473
Message 安全和凭据	473
身份标识管理	474
整体策略	474
场景驱动方法	475
Intranet 应用场景	476
保护 Intranet 绑定	476
约束消息保护	482
身份验证	484
身份标识	486
安全调用上下文	487
模拟	489
授权	496

身份标识管理.....	501
回调.....	502
Internet 应用场景.....	503
保护 Internet 绑定.....	503
消息保护.....	505
验证.....	509
使用 Windows 凭据.....	511
使用 ASP.NET provider.....	512
标识管理.....	520
B2B 应用场景.....	520
保护 B2B 绑定.....	521
验证.....	521
授权.....	524
身份标识管理.....	525
宿主安全配置.....	525
匿名应用场景.....	526
确保匿名绑定的安全.....	526
验证.....	527
授权.....	527
标识管理.....	527
回调.....	528
无安全场景.....	528
无安全的绑定.....	528
验证.....	528
授权.....	529
标识管理.....	529
回调.....	529
场景总结.....	529
声明式安全框架.....	530
SecurityBehaviorAttribute.....	530
宿主端的声明式安全.....	538
客户端的声明式安全.....	539
安全审计.....	546
配置安全审计.....	547
声明式安全审计.....	549
第 11 章 服务总线.....	551
什么是中继服务.....	552
Windows Azure AppFabric 服务总线.....	552
服务总线编程.....	553

中继服务地址.....	553
服务总线注册.....	556
服务总线浏览器.....	558
服务总线绑定.....	558
TCP 中继绑定.....	559
WS 2007 中继绑定.....	563
单向中继绑定.....	563
事件中继绑定.....	564
使用云作为拦截器.....	565
服务总线缓存.....	566
缓存 VS 队列.....	567
使用缓存.....	567
发送和检索消息.....	573
缓存服务.....	574
应答服务.....	582
服务总线验证.....	586
配置验证.....	587
共享安全验证.....	588
无验证.....	592
在服务总线上的元数据.....	594
传输安全.....	596
Transport 安全.....	597
Message 安全.....	597
中继绑定与传输安全.....	598
WS 中继绑定与传输安全.....	604
单向中继绑定与传输安全.....	604
绑定与传输模式.....	605
简化传输安全.....	605
附录 A 面向服务概述.....	613
附录 B 消息头与上下文.....	625
附录 C 服务发现.....	647
附录 D 发布-订阅服务.....	695
附录 E 通用拦截器.....	727
附录 F WCF 编码规范.....	743
附录 G ServiceModelEx 分类.....	753
索引.....	765

本章介绍了WCF的基本概念、构建模块和架构，并指导读者如何构建一个简单的WCF服务。你可以学习到地址（Address）、绑定（Binding）、契约（Contract）和总结点（Endpoint）等基本术语，了解如何托管服务和编写客户端代码，还可以学习与WCF有关的主题，如进程内托管（In-proc hosting）、可靠性（Reliability）、传输会话（Transport session），以及如何在 Visual Studio 2010使用WCF。如果你已经熟悉了WCF相关概念，那么我建议你先快速浏览一下本章的内容，这不仅能巩固已有知识，而且还可以了解一些辅助类和技术术语，以便更好地阅读本书。

什么是 WCF

What Is WCF?

WCF是基于Windows平台下开发和部署服务的SDK。我将会在接下来的章节中介绍什么是服务。但是WCF不只这些——从字面上说，它是更好的.NET。WCF为你的服务提供了运行环境，而且允许你使用CLR类型调用其他服务。理论上，你可以不使用WCF构建服务，但实际上，使用WCF构建服务则更加方便。WCF是微软对一系列行业标准规范的框架实现，这个规范包括服务交互、类型转换、封送，以及各种协议的管理的定义。因此，WCF提供了服务之间的互操作性。

WCF为开发者提供了大多数应用程序都需要的基础功能模块，因此提高了开发效率。WCF的第一个版本（属于.NET 3.0的一部分）为服务开发提供了许多功能，如托管（Hosting）、服务实例管理（Service Instance Management）、异步调用、可靠性、事务管理、断开队列调用（Disconnected Queued Call）及安全性。WCF的第二个版本（属于.NET 3.5的一部分）提供了附加的工具，并在原有的基础上进行了扩展，增加了额外的通信选项。WCF的第三个版本（属于.NET 4.0的一部分）包含了配置变化、一些扩展和新的特性，如服务发现（在附录C中）及路由器（不在本书中讨论）。虽然与.NET 4.0没有直接关系，但是WCF也扩展了对于Windows Azure Platform AppFabricService Bus的支持，这个主题在第11章里作介绍。

此外，WCF同时提供了设计优雅的可扩展模型，使得开发人员能够丰富WCF的基础功能。事实上，WCF自身的实现正是利用了这个可扩展模型。本书的其他章节会致力于探索这些方面的内容和特征。

WCF属于.NET 4.0的一部分，因此它只能运行在支持它的操作系统上。这些操作系统包括Windows XP SP2、Windows Server 2003 SP1、Windows Vista、Windows Server 2008和Windows 7等系统及其更新的版本。

WCF的绝大部分功能都包含在一个单独的程序集 *System.ServiceModel.dll* 中，命名空间是 *System.ServiceModel*。

服务

Service

服务是一组公开的功能集合。从软件设计的角度考虑，软件设计思想经历了从函数发展到对象，从对象发展到组件，从组件发展到服务等几次变迁。在这样漫长的发展过程中，从组件发展到服务的这一步可以说是最具革命意义的一次飞跃。面向服务（*Service-Oriented, SO*）是一组原则的抽象，是创建面向服务应用程序的最佳实践。可以参考附录A，它介绍了使用面向服务的概况和目的。本书其余的内容假设了你对本书的这些原则已经十分了解。面向服务的应用程序（*SOA*）将众多服务聚集到一个单一逻辑的应用程序中，这就类似于面向组件的应用程序聚合组件，或者面向对象的应用程序聚合对象，如图1-1所示。

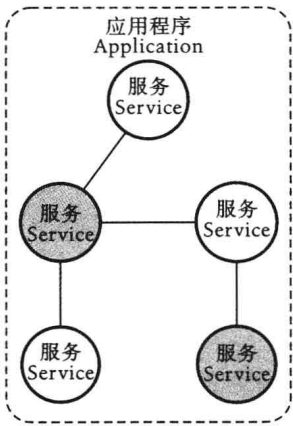


图 1-1: 面向服务的应用程序

服务可以是本地的也可以是远程的，可以由多个用任意技术进行开发的个体开发而成。服务既可以在版本间相互独立，也可以在不同的时区同时进行。服务内部包含了如语言、技术、平台、版本与框架等诸多概念，服务之间的交互则只允

许使用规定的通信模式。

客户端只是使用服务功能的一方。理论上，客户端可以是任意事物，例如，Windows 窗体类、WPF或者是Silverlight类、ASP.NET页面或者其他服务。

客户端与服务通过发送与接收消息进行交互。消息可以直接在客户端与服务之间进行传递，也可以通过中间方式进行传递，如通过Windows Azure AppFabric服务总线。WCF中的消息通常是SOAP消息，与服务不同的是，这些消息与传输协议无关。所以，WCF服务可以在不同的协议之间传输，而不仅限于HTTP。WCF客户端可以与非WCF服务完成互操作，而WCF服务也可以与非WCF客户端交互。因此，如果需要同时开发客户端和服务，则创建的应用程序两端都要求支持WCF，这样才能利用WCF的特有优势。

因为服务的创建对外界是不透明的，也就是说，外界客户端并不知道服务的内部实现细节，所以WCF服务通常通过公开元数据的方式描述可用的功能及服务可能采用的通信方式。元数据的发布可以预先定义，它与具体的技术无关（*Technology-Neutral*），如采用基于HTTP-GET

方式的WSDL或符合元数据的任何交换协议。一个非WCF客户端可以将元数据作为本地类型导入本地的环境中。相似地，WCF客户端也可以导入非WCF服务的元数据，然后以本地CLR类和接口的方式进行调用。

服务的执行边界

WCF不允许客户端直接与服务交互，即使它调用的是本地机器内存中的服务也不允许。相反，客户端总是使用代理（proxy）将调用转发给服务。代理公开的操作与服务相同，同时还增加了一些管理代理的方法。

WCF允许客户端跨越执行边界和服务通信。在同一台机器中，客户端可以调用同一个应用程序域中的服务，也可以在同一进程中跨应用程序域调用，甚至跨进程调用，如图1-2所示。

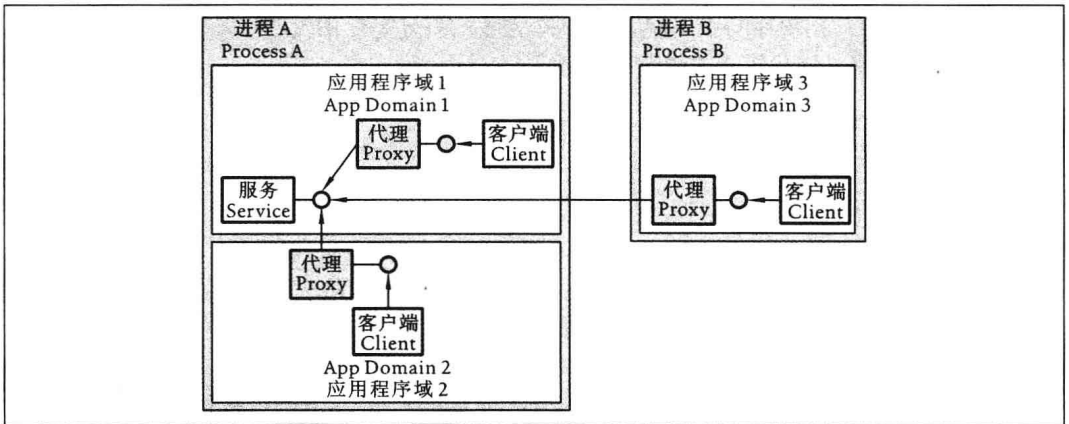


图 1-2：使用 WCF 在同一机器上通信

跨越机器边界通信（见图1-3）是指客户端可以与服务跨越Internet或者Intranet交互。

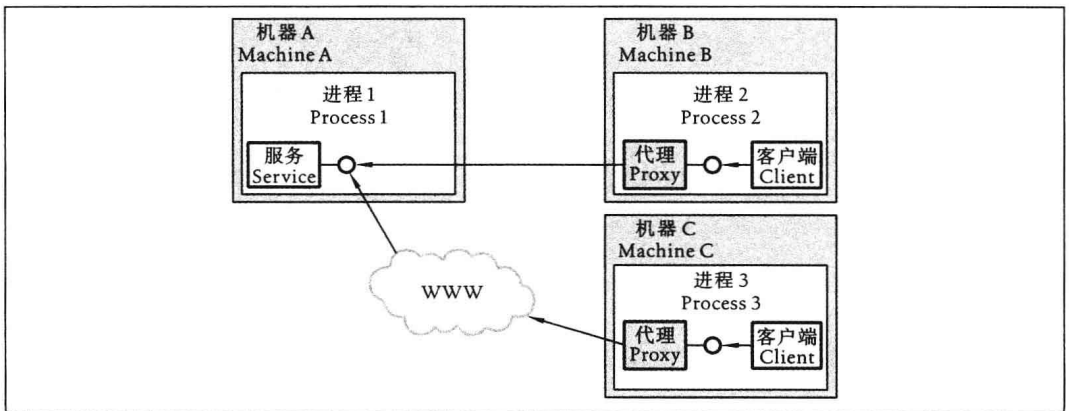


图 1-3：使用 WCF 实现跨越机器通信