



云计算架构 设计模式

Alex Homer
John Sharp
[美] Larry Brader 著
Masashi Narumoto
Trent Swanson

新青年架构小组 译



华中科技大学出版社

<http://www.hustp.com>

patterns & practices

云计算架构设计模式

Alex Homer
John Sharp
[美] *Larry Brader* 著
Masashi Narumoto
Trent Swanson
新青年架构小组 译

华中科技大学出版社

中国·武汉

图书在版编目(CIP)数据

云计算架构设计模式/(美)艾利克斯·洪木尔(Alex Homer)等著;新青年架构小组译. —武汉:华中科技大学出版社,2017.10

ISBN 978-7-5680-3402-9

I. ①云… II. ①艾… ②新… III. ①云计算-架构 IV. ①TP393.027

中国版本图书馆 CIP 数据核字(2017)第 251567 号

云计算架构设计模式

Yunjisuan Jiagou Sheji Moshi

[美] Alex Homer, John Sharp, Larry Brader, 著
Masashi Narumoto, Trent Swanson
新青年架构小组 译

策划编辑: 谢燕群

责任编辑: 谢燕群

责任校对: 何 欢

责任监印: 周治超

出版发行: 华中科技大学出版社(中国·武汉)

电话: (027)81321913

武汉市东湖新技术开发区华工科技园

邮编: 430223

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×960mm 1/16

印 张: 15.25

字 数: 387 千字

版 次: 2017 年 10 月第 1 版第 1 次印刷

定 价: 48.00 元



华中出版

本书若有印装质量问题, 请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

译者序

非常荣幸，能够获得我一直非常崇拜的计算团队——微软模式与实践小组(微软P&P小组，Microsoft Patterns & Practices Group)的授权，翻译这部经典著作。

微软P&P小组影响了我10多年的技术生涯，这是我最喜欢的微软技术小组，偶像团队！低调、谦虚、务实！是最早从事开源项目的微软技术团队之一，坚持理论和实践相结合，把最经典的软件工程理论知识用到具体的项目开发中。如果你是Java程序员，也不要介意，我相信你非常喜欢Spring开发团队，而微软P&P小组在.NET领域的地位与其相当。

架构即未来

移动互联网、云计算、大数据时代，面临更多的技术挑战问题，设计模式已经从单一的OO问题领域向外扩张延伸，设计模式的范畴不会局限于语言本身，更多扩展到架构设计领域。

我个人对编程语言没有偏见，它们各有所长，大家互相学习。有实力的程序员不需要通过编程语言来找优越感，都在踏实地研究技术。我接触的优秀程序员无论是对C++、Java，还是对C#等都没有编程语言的歧视。况且底层的算法、数据结构和设计模式都没有编程语言限制。

架构师和技术专家是程序员中的精英群体、技术领袖，在公司也是受人尊敬的群体。

十年磨一剑 精华知识

本书介绍了云计算时代最新、最经典的24种架构设计模式，包含基于云平台设计架构面临的问题以及典型的解决方案，另外还有10个架构设计指南。

24种经典的设计模式包括：高并发、健康健康、消息编排、架构伸缩、缓存、消息推送、大数据存储和优化、安全令牌等架构设计的关键问题，是一本最近几年难得的架构与设计模式图书。

本书内容不受编程语言限制，可以根据需要使用.NET、Java、PHP、Node.js、Go等。云计算平台可以是Azure、AWS、阿里云等共有云，也可以是私有云平台。

推荐本书给高级工程师、运维工程师、架构师、技术经理、技术总监、CTO学习使用。

微软 P&P 模式与实践小组

微软P&P小组鼎鼎的大名是微软模式与实践小组亲自打造的，为开发基于云计算架构应用程序遇到的常见问题提供了经典的解决方案，并将常用的经典方案归类为设计模式。

微软P&P模式与实践小组是微软最早的开源社区团队之一，主要关注于把行业经典设计模式与实际项目开发相结合的技术研究。

微软早期众多的开源项目都是由该小组主导完成，同时还编写了许多经典书籍和代码。

如果你坚持研究.NET技术10年以上，就一定看过微软P&P小组的学习资料。

在.NET领域工作10年以上的程序员应该不会对P&P小组陌生。其技术实力非常强，国内很多.NET架构师都是看P&P小组的文章和代码成长起来的。

我在“菜鸟”阶段，第一次看完P&P小组的文章就成为其忠实的粉丝，至今依然向技术圈子的朋友和新青年架构班的同学推荐他们的资料。

10年前P&P小组开源的Enterprise Library代码、分布式与安全的文档，到后来的IOC容器Unity等，都是精华知识的沉淀。

23 种设计模式

Erich Gamma等在《设计模式》一书种介绍了23种经典的设计模式。

本书介绍的24种经典设计模式包含高并发、健康健康、消息编排、架构伸缩、缓存、消息推送、大数据存储和优化、安全令牌等架构设计的关键问题，是难得的最新的设计模式图书，可以作为最经典的Erich Gamma、Richard Helm、Ralph Johnson、John Vlissides 4大金刚的《设计模式》的扩展阅读资料。

24 种云计算架构模式

本书介绍的24种经典设计模式包含高并发、健康健康、消息编排、架构伸缩、缓存、消息推送、大数据存储和优化、安全令牌等架构设计的关键问题，是难得的最新设计模式图书。

- | | |
|----------------|---------------------|
| (1) 缓存驻留模式 | (2) 断路器模式 |
| (3) 事务补偿模式 | (4) 竞争消费者模式 |
| (5) 计算资源合并模式 | (6) 命令和职责分离(CQRS)模式 |
| (7) 事件溯源模式 | (8) 外部配置存储模式 |
| (9) 联合身份模式 | (10) 门卫模式 |
| (11) 健康终结点监控模式 | (12) 索引表模式 |

- (13) 领导选举模式
- (14) 物化视图模型
- (15) 管道和过滤器模式
- (16) 优先级队列模式
- (17) 基于队列的负载均衡模式
- (18) 重试模式
- (19) 运行时重配置模式
- (20) 调度器代理监控模式
- (21) 分片模式
- (22) 静态内容托管模式
- (23) 限流模式
- (24) 令牌模式

24种架构设计模式都有对应的例子代码，可以下载参考。Java程序员也可以从中获益良多。

10 个架构设计指南

除了详细介绍了24种云计算架构设计模式以外，这里还介绍了实际架构设计中的重要原则，归类为10个方面，可作为大家进行架构设计时的参考。

- (1) 异步消息传输
- (2) 自动伸缩指南
- (3) 缓存指南
- (4) 计算分区指南
- (5) 数据一致性指南
- (6) 数据分区指南
- (7) 数据复制与同步指南
- (8) 远程监控指南
- (9) 多数据中心部署指南
- (10) 服务调用统计指南

中文授权翻译声明

本书翻译获得了微软模式与实践小组(微软P&P小组)的中文翻译授权。本着学习的原则，翻译并分享本书。感谢微软模式与实践小组的认可。感谢微软中国的支持。

我们翻译本书时采用人工翻译、手工打字、人工校验，由新青年技术架构小组协作完成。按翻译顺序的译者排名如下：

徐雷	于玉爽	熊旭	段晓婵	杨海伦	包欣雪	Fiona
秦孝文	蓝猴子	小武	成松	任王江	韩庆瑞	孙聪
王永乐	天涯	张刚	吴亦双	安晓磊	李文东	文君
郑承良	孙强	庄思捷	叶祖豪	曾真理	徐扬	申浩

徐雷FrankXuLei
2017年10月8日

关于作者

About the authors

Alex Homer是微软模式与实践小组的技术写作者。在加入微软之前，他做了多年的软件设计和培训工作。他在小组的主要精力花费在设计模式和架构方面，还编写指南和例子代码。他的博客地址是<http://blogs.msdn.com/alexhomer/>。

John Sharp 是Content Master (www.contentmaster.com)的首席技术专家。专注于使用.NET框架和Azure平台开发应用系统。John写过《Microsoft Visual C# Step by Step》和《Microsoft WCF Step by Step》。John讲授过许多培训课程，也写过许多覆盖诸多领域，比如C和C++编程、SQL Server数据库管理以及面向服务架构的技术文章。他获得了伦敦大学帝国理工学院的计算机荣誉学位。

Larry Brader是微软模式与实践小组的高级测试工程师，负责小组不同项目的测试工作，专注于客户端和服务端。此外他对基于ALM生成测试指南也有浓厚的兴趣。

Masashi Narumoto热衷于将互联网作为知识库的想法。互联网已经显著地改变了我们的生活，毫无疑问还会带来巨大的变化。他的目标是汇聚大家的智慧以更有意义的形式提供出来，方便更多的人学习。在就职于微软模式与实践小组期间，他作为程序经理负责完成了多个Azure指南的系列文档的编写工作，现在关注于大数据领域。之前，他花费了20多年时间来开发和咨询各种不同的解决方案，尤其是零售和制造业。Masashi 的博客地址是http://blogs.msdn.com/masashi_narumoto，推特账号是@dragon119。

Trent Swanson是Full Scale 180的软件架构师，也是创始人之一，主要使用云计算技术。他一开始就使用Azure技术，帮助全球各地的客户构建、部署和管理Azure上的云计算解决方案。无论是迁移现有应用到云计算平台还是构建全新应用，他都享有整个交付伸缩的、可靠的和可管理的云计算解决方案专利。

序言

Preface

本书由微软模式与实践小组（微软P&P小组）亲自打造，得到了许多社区开发者的支持，为开发基于云计算架构的应用程序常见问题提供了解决方案。

本书指南

- 介绍实现云计算应用时，尤其是托管在Azure云平台时使用这些设计模式的好处。
- 讨论云计算设计模式的经典问题和解决方案，以及它们如何与Azure关联到一起。
- 展示如何使用Azure功能实现这些模式，强调其优点与顾虑。
- 通过描述如何把这些设计模式应用到云计算应用架构中以及它们之间的关系来描绘知识体系的宏伟蓝图。

本书介绍的主题适用于所有的分布式系统，无论是托管在Azure中还是其他云计算平台中。

我们的目的并非提供设计模式的详尽集合，而是选择了对云计算应用最有用处的设计模式——尤其考虑到在用户中的流行度。本书不是详细介绍Azure平台功能的指南。要学习Azure可以参阅<http://azure.com>。

本书内容

结合开发社区的代表性反馈，我们把云计算应用开发领域最常见的问题归纳为8类。

类别	描述
可用性	可用性定义为系统正常工作的时间比例。它受系统错误、基础架构问题、各种工具以及系统负载的影响。通常根据系统正常运行的时间来衡量。云计算应用为用户提供了服务级别协议(SLA)，它指的是应用程序必须以最大化可用性的方式设计和实现
数据管理	数据管理是云应用的关键部分，并且影响质量特性。由于诸如性能、伸缩性或者可用性等原因，数据通常存储在不同服务器的不同位置上，这些也会带来新的挑战。例如，数据一致性必须是可维护的，并且数据需要进行跨区域同步

类别	描述
设计和实现	良好的设计会在组件设计和部署里包含诸如一致性和内聚性、简化管理和部署的可维护性，以及允许组件和子系统被其他应用使用的重用性。设计和实现阶段的决策对于云计算应用和服务的总体质量及成本有重大影响
消息	云计算应用的分布式特性需要一个连接组件和服务的消息基础架构，理想情况下是松耦合方式，便于最大化伸缩性。异步消息被广泛使用，而且提供了许多好处，同时也带来了许多挑战，比如消息顺序、毒消息管理、幂等性等
管理和监控	云计算应用运行在远程数据中心中，我们无法完全控制基础架构或者操作系统。相比私有云，这种情况使得管理工作更加困难。只有应用暴露运行时的信息，管理员和运营人员才可以管理和监控系统；只有支持修改业务需求和自定义，才不需要应用停止和创新部署
性能和伸缩性	性能是系统执行特性操作的响应性指标，而伸缩性是系统处理新增压力但不会影响性能与可用性的能力。云计算应用通常会遇到变化的工作负载和峰值，这种情况通常不可预测，特别是在多租户场景下。相反，应用应该能够通过伸缩来满足峰值的需要，而且当需要下降时回退。伸缩性不只是关注计算实例，还关注其他要素，比如数据存储、消息基础架构等
弹性	弹性是系统优雅地处理错误和恢复系统的能力。云计算主机的基本特性为：应用通常是多租户的，使用共享平台服务，竞争资源和带宽，通过互联网通信，运行在商用硬件上。这意味着将会出现更多短暂和永久的错误。探测错误，并快速和高效地恢复对于维护弹性来说必不可少
安全	安全是系统阻止设计使用范围外的恶意和意外操作的能力，是阻止泄露和丢失数据的能力。云计算应用暴露在互联网上，跨越信任的私有云环境边界，通常对外开放，因此可能会出现不信任的用户。应用必须以安全的方式设计和部署，避免恶意攻击，限制只有支持的用户可以访问，并且保护敏感数据

对于每个类别，我们都创建了关联的指南和文档，以帮助开发者解决常见的共同问题。包括以下3方面内容。

- 24种设计模式。这是云托管应用非常有用的24种设计模式。每种设计模式提供了描述上下文及其问题，解决方案及其问题，使用模式的公共格式，以及基于Azure平台的例子。每种模式也包含与其他相关模式的链接。
- 10个指南主题。提供了开发云计算应用所需的基本知识、实践经验与技巧。每个主题都详细地介绍了这些知识。
- 例子程序。演示了设计模式的使用过程。我们可以使用这些代码参考设计自己特定的需求代码。

设计模式

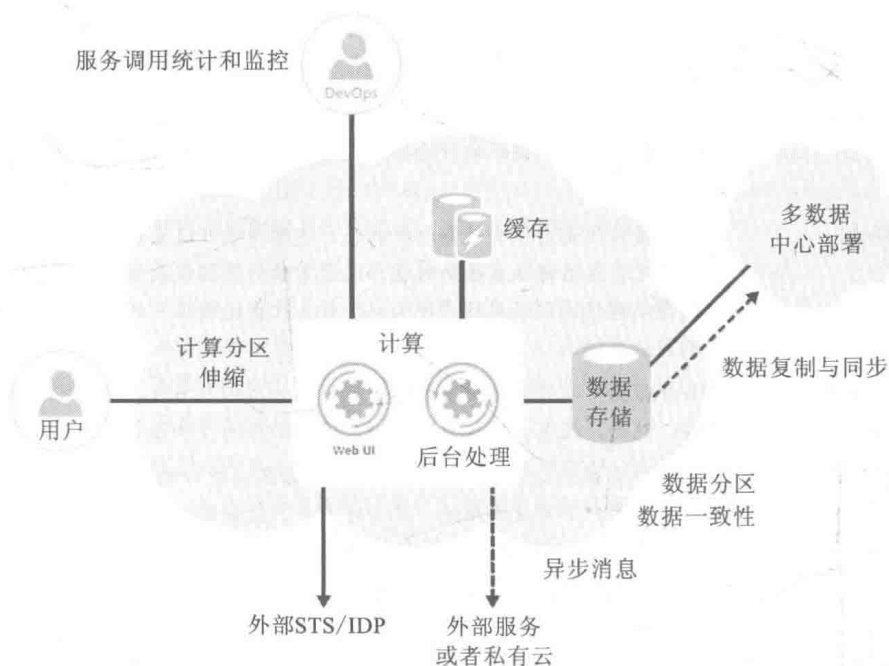
设计模式被分配到一个或者多个类别中。完整的设计模式列表如下：

模式	描述
缓存驻留模式	根据需要从数据存储器加载数据。此模式可以用来改进性能，还可以用于维护缓存和后台数据库之间的数据一致性
断路器模式	当连接远程服务或资源时，可能导致不定时间恢复系统的错误。此模式可以用于改进系统的稳定性和弹性
事务补偿模式	如果一个或者多个操作失败，就会取消执行的一系列工作，它定义了一组最终一致性模型的操作。遵从最终一致性模型的操作在云托管应用中非常常见，通常都会实现复杂的业务过程和工作流
竞争消费者模式	允许多个并发的消费者在相同的消息通道上处理接收的消息。此模式允许系统并发处理多个消息以优化系统的吞吐量，改进系统的伸缩性和可用性，平衡工作负载
计算资源合并模式	合并多个任务或者操作到单个计算单元里。此模式可以增加计算资源的使用率，降低云计算应用中计算处理的成本和管理开销
命令和职责分离 (CQRS) 模式	通过隔离接口来分离更新和读操作。此模式可以最大化性能、伸缩性和安全性，通过高度的灵活性支持系统的进化，在领域级别阻止引起合并冲突的更新命令
事件溯源模式	使用只能追加的存储库来记录领域里发生在数据库方面的完整操作事件序列，而不是仅仅保存当前状态，让存储库可以用来创建特定的对象状态。此模式可以通过避免同步数据模型和业务模型的需求来简化复杂领域里的任务，改进性能、伸缩性和响应性，提供事务性数据的一致性，维护可以支持补偿操作的完整审计追踪和历史信息
外部配置存储模式	把配置信息从应用部署包移动到一个中心位置。此模式可以提供更简单的管理和配置数据控制、跨应用和应用实例共享配置的机会
联合身份模式	把验证委托给一个外部身份标识提供器。此模式可以简化部署，最小化用户管理的需求，并且改进应用的用户体验
门卫模式	通过在客户端和应用与服务之间使用特定的作为代理的宿主实例来保护应用和服务，验证并保护请求，且在它们之间传递请求消息。此模式可以提供额外的安全层，并且降低系统受攻击的层面
健康终结点监控模式	在应用中实现功能检查，可以通过暴露的终结点定时访问监控数据。此模式可以帮助检验应用和服务是否正确执行
索引表模式	在频繁访问的数据存储库特定字段上出 aung 时就索引。此模式可以通过允许应用更快速地从数据存储库查询数据来改进查询性能

模式	描述
领导选举模式	选举一个实例作为领导来承担管理其他实例的职责，让其协调分布式节点的任务执行。此模式可以帮助确保任务不会与其他任务冲突，避免资源争用，或者被其他执行的任务干扰
物化视图模型	当数据格式并非查询操作期望的格式时，提前为一个或者多个数据存储器中的数据生成视图。此模式可以帮助实现高效查询和数据提取，改进应用程序的性能
管道和过滤器模式	把一个复杂的任务分解为一系列可以单独执行的可重用的任务元素。此模式可以通过独立部署和伸缩任务元素来改进性能、伸缩性和可重用性
优先级队列模式	为发送给服务的请求消息设置优先级，这样高优先级的请求可以被更快地处理。此模式在给独立类型的客户端提供不同服务级别担保时非常有用
基于队列的负载均衡模式	在任务和调用的服务之间使用队列作为缓冲区来平滑断断续续的可能导致服务失败或者任务超时的超量负载。此模式可以帮助最小化峰值压力对于任务和服务的可用性和响应能力的影响
重试模式	当连接服务或者网络资源时，通过允许短暂错误重试操作来允许应用程序处理临时的失败。此模式可以用来改进应用的稳定性
运行时重配置模式	设计应用程序使其可以无需重新部署，或者重新启动应用来重新配置。这可以用于维护可用性和最小化宕机时间
调度器代理监控模式	协调跨分布式服务和其他资源的集合。如果某个操作失败，则尝试透明地处理错误；或者如果系统无法从错误中恢复，则取消执行工作产生的影响。此模式可以通过对短暂异常、长期错误和处理错误启用恢复和重试操作增加系统的弹性
分片模式	把数据库水平分割为不同的区片进行存储。当存储和访问海量数据时，此模式可以改进伸缩性
静态内容托管模式	部署静态内容到可以直接发送给客户端的云端存储服务上。此模式可以减少对于昂贵计算实例的需求
限流模式	控制单个应用实例、单个租户或者整个服务消耗的资源数量。此模式可以允许系统继续工作，并且满足服务级别协议，甚至对资源增加极限负载时也可以正常工作
令牌模式	为了在应用程序代码里支持卸载数据传输操作，使用令牌或者密钥来限制对特定资源或服务的访问。此模式在使用云托管存储系统或者队列时特别有用，并且可以最小化成本，最大化伸缩性和性能

主题

这些主题与特定的应用程序开发相关，如下图所示。



这个指南包含下面的主题。

主题	描述
异步消息通信指南	消息通信是许多分布式系统采用的关键策略，比如云计算。它允许应用和服务彼此通信并协同工作，且可以帮助构建可伸缩的和弹性的解决方案。消息通信支持异步操作，允许我们解耦服务调用和宿主进程
自动化伸缩指南	持续监控性能和伸缩系统以适应波动的工作负载、满足目标需求并且优化运营成本，这个过程可能需要大量的人力成本。这些工作可能无法人工完成。这也是自动化伸缩的用武之地
缓存指南	缓存是一种常见的改进系统性能和伸缩性的技术，它通过把高频率访问的数据复制到接近应用的数据存储区中来实现加速。缓存当应用程序重复读取相同的数据时最高效，特别是原始数据存储区的速度相对缓存较慢的时候，它受制于高级别的争用，或者说它不会导致网络延迟
计算分区指南	当部署应用程序到云端时，很可能采用把服务和组件根据使用情况分别部署的方式，以便在维护伸缩性、性能、可用性和应用安全时最小化运行成本
数据一致性指南	云应用通常使用的数据分散存储在不同的存储区里。管理和维护数据一致性变得尤为重要，尤其是出现并发性和可用性问题时。我们通常需要在并发性和一致性之间取舍。这意味着我们在设计解决方案时会考虑最终一致性，而不会追求应用程序所有时刻都处于完全一致性状态
数据分区指南	在许多大型伸缩解决方案中，数据被分割到不同的可以单独管理和访问的分区中。必须仔细选择这些分区策略以便最小化坏处、最大化好处。分区可以帮助我们改进伸缩性，降低争用，并且优化性能

主题	描述
复制和同步数据指南	当部署应用到多个数据中心，比如云和私有云中时，为了最大化可用性和性能，确保一致性，最小化数据传输成本，就必须考虑如何在多个节点同步数据。
远程监控指南	绝大多数应用都会包含诊断功能特性，这些特性可以生成自定义监控和调试信息，尤其是当错误发生的时候。这通常称为监控仪表盘，是通过添加事件和错误处理代码到应用程序中来实现的。收集远程信息的过程通常称为遥感监测。
多数据中心部署指南	在多个数据中心部署应用有许多好处，比如增加可用性、更好的跨地区用户体验。然而，还有一些挑战要解决，比如数据同步和监管限制。
服务调用统计指南	我们可能需要统计应用或者服务的调用信息以便计划、调整未来的需求策略，了解用户如何使用系统，或者对用户、组织部门或者客户计费。这些都是常见的需求，对大公司和独立软件供应商及服务商尤其如此。

例子应用

10个例子程序演示了本书中介绍的一些模式实现过程，大家可以下载到本地运行，或者部署到Azure订阅账号里测试。要获取并运行这些程序代码，则

- (1) 可以到微软下载中心<http://aka.ms/cloud-design-patterns-sample> “Cloud Design Patterns - Sample Code”页面下载 “Cloud Design Patterns Examples.zip”压缩文件。
- (2) 在Windows资源浏览器里打开压缩文件的属性，选择解压。
- (3) 把代码复制到磁盘根目录，比如C:\PatternsGuide。不要在用户配置文件夹里解压(比如Documents 或者 Downloads)，否则可能导致文件名过长。
- (4) 在浏览器里打开Readme.htm。它包含系统和例子的配置信息，在本地Azure模拟器运行例子或者部署到Azure平台上，掌握例子展示的知识点。

本指南与例子的对应关系如下表所示。

主题	描述
竞争消费者	这个例子包含 2 个组件：Sender worker role 负责发送消息到 Service Bus 队列，Receiver worker role 负责从队列接收消息并处理消息。Receiver worker role 启动两个实例来模拟消费者竞争。
计算资源合并	这个例子展示了如何统一多个独立的任务到单个的 worker role 中。运行这个例子没有其他需求。

主题	描述
外部配置存储	这个例子展示了在外部存储区保存配置文件而不是使用本地配置文件。在这个例子中，配置保存到 Azure Blob Storage 存储区。Blob 包含的配置信息是通过 ExternalConfigurationManager 类的实例监控的。当 ExternalConfigurationManager 对象探测到配置已修改时，它就会提醒应用程序
健康终结点监控	这个例子展示了如何设置检查独立服务健康状态的 Web 终结点，它可以返回不同的状态码来表示状态。设计终结点的目的是让看门狗监控服务监控，比如 Azure 终结点健康监控服务，但是我们也可以在浏览器里打开和调用终结点，查看状态结果。我们还可以部署和配置自己的终结点健康工具来发送请求给服务操作并分析接收到的应答消息
领导选举	这个例子展示了工作角色实例如何变成领导。领导可以协调和控制其他实例的任务，这些任务应该通过某个工作角色实例执行。领导选举通过获取租赁权来实现
管道和过滤器	这个例子包含两个可以执行整体处理部分操作的过滤器。这两个过滤器组合在一个管道中，一个过滤器的输出结果作为另一个过滤器的输入数据。过滤器作为单独的工作角色实现，Azure Service Bus 总线队列提供了管道的基础架构
优先级队列	这个例子展示了如何通过 Service Bus 主题和订阅实现优先级队列。工作角色负责发送消息给主题，分配优先级给每个消息。接收工作角色从对应优先级订阅者中读取消息。在这个例子中，PriorityQueue.High 工作角色运行两个实例，PriorityQueue.Low 只运行一个实例。这样就能确保高优先级消息可以更快地被读取
运行时重配置	这个例子展示了如何修改云服务的配置而不需要重新启动 Web 和 worker role
静态内容托管	这个例子展示了如何从公共存储服务中快速获取静态内容。这个例子包含了一个 Azure Web Role，它托管了包含 JavaScript 文件和图片的网站，部署到 Azure 存储区中。这些内容通常部署到存储账号中，作为应用部署的一部分。但是，为了简化例子，启动程序时这些文件会被部署到存储账号中
令牌模式	这个例子展示了客户端程序如何在获取一个带有权限的共享访问前直接向大对象存储区写入数据。为了简单明了，这个例子关注获取和消费令牌密钥的机制，不会展示如何实现验证或者安全通行

这些例子关注演示每个模式的关键功能特性，并不可以直接使用到生产环境。

更多信息

所有的章节都包含对于其他资源的参考，比如图书、博客文章以及论文。如果需要，大家可以阅读更详细的内容。为了方便大家，这里有一个网页包含所有链接的文章，直接点击 <http://aka.ms/cdpbibliography>，大家可以阅读、查询资源。

反馈和支持

问题？评论？建议？任何关于本书的反馈或者要获取任何问题的帮助，请访问 <http://wag.codeplex.com>。这个社区网站的论坛是我们推荐的反馈和支持渠道，因为它允许我们大家分享想法、问题和整个社区的解决方案。

图书团队

创意与项目管理: Masashi Narumoto

作者: Alex Homer、John Sharp、Larry Brader、Masashi Narumoto和Trent Swanson

开发: Julian Dominguez、Trent Swanson (Full Scale 180)、Alejandro Jezierski (Southworks)

测试: Larry Brader、Federico Boerr和Mariano Grande (Digit Factory)

性能测试: Carlos Farre、Naveen Pitipornvivat (Adecco)

文档: Alex Homer、John Sharp (Content Master Ltd)

图片艺术家: Chris Burns (Linda Werner & Associates Inc)、Kieran Phelan (Allovus Design Inc)

编辑: RoAnn Corbisier

生产: Nelly Delgado

技术审阅: Bill Wilder (Author, Cloud Architecture Patterns)、Michael Wood (Cerebrata)

贡献者: Hatay Tuna、Chris Clayton、Amit Srivastava、Jason Wescott、Clemens Vasters、Abhishek Lal、Vittorio Bertocci、Boris Scholl、Conor Cunningham、Stuart Ozer、Paolo Salvatori、Shirley Wang、Saurabh Pant、Ben Ridgway、Rahul Rai、Jeremiah Talkar、Simon Gurevich、Haishi Bai、Larry Franks、Grigori Melnik、Mani Subramanian、Rohit Sharma、Christopher Bennage、Andrew Oakley、Jane Sinyagina、Julian Dominguez、Fernando Simonazzi (Clarius Consulting)和Valery Mizonov (Full Scale 180)

微软开发者指南顾问委员会暨参与审核成员名单: Carlos dos Santos、CDS Informatica Ltda; Catalin Gheorghiu、I Computer Solutions; Neil Mackenzie、Satory Global; Christopher Maneu、Deezer.com; Paulo Morgado; Bill Wagner、Bill Wagner Software LLC; Roger Whitehead、ProSource.It

感谢为本书的顺利出版付出心血的每个参与者！

目录

Table of Contents

第1章 缓存驻留模式.....	1
背景和问题.....	1
解决方案.....	1
相关模式与指南.....	5
更多信息.....	5
第2章 断路器模式.....	6
背景和问题.....	6
解决方案.....	6
问题与思考.....	8
何时使用此模式.....	10
相关模式和指南.....	14
第3章 补偿事务模式.....	15
背景和问题.....	15
解决方案.....	16
问题与思考.....	16
何时使用此模式.....	17
相关模式和指南.....	18
第4章 竞争消费者模式.....	19
背景和问题.....	19
解决方案.....	19
问题与思考.....	20
何时使用此模式.....	21
第5章 计算资源合并模式.....	25
背景和问题.....	25
解决方案.....	26
问题与思考.....	26
应用场景.....	28
相关模式和指南.....	32
扩展阅读.....	32

第6章 命令和查询职责分离(CQRS)模式.....	33
背景和问题.....	33
解决方案.....	34
问题与思考.....	35
何时使用此模式.....	36
第7章 事件溯源模式.....	41
背景和问题.....	41
解决方案.....	41
问题与思考.....	43
何时使用此模式.....	44
相关模式和指南.....	47
第8章 外部配置存储模式.....	48
背景和问题.....	48
解决方案.....	48
问题与思考.....	49
何时使用此模式.....	50
关联模式和指南.....	56
第9章 联合身份模式.....	57
背景和问题.....	57
解决方案.....	57
问题与思考.....	59
何时使用此模式.....	59
相关模式和指南.....	61
更多信息.....	61
第10章 门卫模式.....	62
背景和问题.....	62
解决方案.....	62
问题与思考.....	63
何时使用此模式.....	63
相关模式和指南.....	64
第11章 健康终端监控模式.....	65
背景和问题.....	65
解决方案.....	65
问题与思考.....	67
何时使用此模式.....	68
相关模式和指南.....	71
更多信息.....	71