

<<设计模式>>

图书基本信息

书名：<<设计模式>>

13位ISBN编号：9787302279976

10位ISBN编号：7302279977

出版时间：2012-4

出版时间：清华大学出版社

作者：（美）坎贝尔 等著，邹雪梅，李岸 译

页数：214

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<设计模式>>

内容概要

《设计模式：NET并行编程》结合大量的项目实践，介绍了与并行编程相关的概念、方法和应用。本书共7章：第1章主要介绍并行编程的基本概念与并行计算的基础理论，第2章主要介绍并行循环的知识，第3章介绍并行任务处理，第4章阐述并行合并计算的机理，第5章介绍future模式，第6章在前文的基础上深入探讨动态并行任务机制，第7章介绍并行编程的流水线机制。

《设计模式：NET并行编程》适用于在 .netframework 上编写托管代码的程序员，包括在 visualc#、visualbasic 以及 visualf# 上编写代码的程序员。本书不假定读者具有并行编程技术的预备知识。不过，读者需要熟悉 c# 的特征，如委托、lambda 表达式、泛型以及语言集成查询（linq）表达式等。读者还至少应该对进程和线程的概念有基本的了解。

<<设计模式>>

作者简介

作者：(美) Colin Campbell Ralph Johnson Ade Miller Stephen Toub 译者：曹泽文 邹雪梅 李岸 Colin Campbell是Model—Based Software Testing and Analysis in C#的合著者之一，他发表过数篇有关软件分析的论文。

他是西雅图的Modeled Computation LLC的创始人和负责人。

Ralph Johnson是伊利诺伊大学的研究副教授。

他是Design Patterns的四个合著者之一，并且是开发了第一个自动重构工具Smalltalk Refactoring Browser的项目组的组长。

近几年来，他一直致力于记载并行编程的模式。

Ade Miller是微软的patterns & practices组的主力开发，他在这里管理了数个敏捷团队，这些团队为微软的客户提供项目上的实用指导。

他的首要兴趣在于并行计算和敏捷软件开发实践。

Stephen Toub在微软的并行计算平台团队中工作。

他致力于为.NET和Visual Studio设计和开发下一代并发和并行编程模型。

<<设计模式>>

书籍目录

第1章 导论

- 1.1 潜在并行性的重要性
- 1.2 分解、伪、调和可扩展共享
 - 1.2.1 了解任务
 - 1.2.2 协调任务
 - 1.2.3 数据的可扩展共享
 - 1.2.4 设计方法
- 1.3 选择恰当的模式
- 1.4 关于术语
- 1.5 并行性的极限
- 1.6 一些技巧
- 1.7 练习
- 1.8 扩展阅读

第2章 并行循环

- 2.1 基础知识
 - 2.1.1 并行for循环
 - 2.1.2 并行foreach循环
 - 2.1.3 并行linq (plinq)
 - 2.1.4 预期
- 2.2 示例
 - 2.2.1 信贷审查的顺序版本示例
 - 2.2.2 使用parallel.foreach的信贷审查示例
 - 2.2.3 plinq信贷审查示例
 - 2.2.4 性能比较
- 2.3 变化形式
 - 2.3.1 尽早中断循环
 - 2.3.2 外部循环取消
 - 2.3.3 异常处理
 - 2.3.4 小循环体的特殊处理
 - 2.3.5 控制并行度
 - 2.3.6 在循环体中使用局部任务状态
 - 2.3.7 对并行循环使用自定义的任务调度程序
- 2.4 反模式
 - 2.4.1 步长不为1
 - 2.4.2 隐藏的循环体依赖
 - 2.4.3 少量迭代的小循环体
 - 2.4.4 处理器的超额申请和申请不足
 - 2.4.5 混合parallel类和plinq
 - 2.4.6 输入枚举中的重复
- 2.5 设计说明
 - 2.5.1 自适应分区
 - 2.5.2 自适应并发
 - 2.5.3 支持嵌套循环和服务器应用程序
- 2.6 相关模式
- 2.7 练习

<<设计模式>>

2.8 扩展阅读

第3章 并行任务

3.1 基础知识

3.2 示例

3.3 变化形式

3.3.1 取消任务

3.3.2 处理异常

3.3.3 等待第一个任务完成

3.3.4 推测执行

3.3.5 使用自定义的调度方式创建任务

3.4 反模式

3.4.1 闭包捕获的变量

3.4.2 清理任务所需要的资源

3.4.3 避免撤销线程

3.5 设计说明

3.5.1 任务和线程

3.5.2 任务生命周期

3.5.3 编写自定义的任务调度程序

3.5.4 未观测到的任务异常

3.5.5 数据并行性和任务并行性之间的关系

3.6 默认任务调度程序

3.6.1 线程池

3.6.2 分散管理的调度技术

3.6.3 workstealing策略

3.6.4 全局队列中的顶层任务

3.6.5 局部队列中的子任务

3.6.6 子任务的内联执行

3.6.7 线程注入

3.6.8 绕过线程池

3.7 练习

3.8 扩展阅读

第4章 并行合并计算

4.1 基础知识

4.2 示例

4.3 变化形式

4.3.1 使用并行循环进行合并计算

4.3.2 使用范围分割器进行合并计算

4.3.3 使用带有范围选择的plinq合并计算

4.4 设计说明

4.5 相关的模式

4.6 练习

4.7 扩展阅读

第5章 future模式

5.1 基础知识

5.1.1 future

5.1.2 延续任务

5.2 示例：adatum金融仪表盘

<<设计模式>>

- 5.2.1 业务对象
- 5.2.2 分析引擎
- 5.2.3 视图和视图模型
- 5.3 变化形式
 - 5.3.1 取消future和延续任务
 - 5.3.2 拥有多个先行任务的情况
 - 5.3.3 使用.net异步调用和future
 - 5.3.4 消除瓶颈
 - 5.3.5 运行时修改图
- 5.4 设计说明
 - 5.4.1 分解成future和延续任务
 - 5.4.2 函数式风格
- 5.5 相关的模式
 - 5.5.1 流水线模式
 - 5.5.2 主 / 从 (master / worker) 模式
 - 5.5.3 动态任务并行模式
 - 5.5.4 离散事件模式
- 5.6 练习
- 5.7 扩展阅读
- 第6章 动态任务并行
 - 6.1 基础
 - 6.2 示例
 - 6.3 变化形式
 - 6.3.1 while-not-empty并行
 - 6.3.2 任务链与父子任务
 - 6.4 设计说明
 - 6.5 练习
 - 6.6 扩展阅读
- 第7章 流水线
 - 7.1 基础
 - 7.2 示例
 - 7.2.1 顺序图像处理
 - 7.2.2 图像流水线
 - 7.2.3 运行特性
 - 7.3 变化形式
 - 7.3.1 取消流水线
 - 7.3.2 处理流水线异常
 - 7.3.3 利用多个生产者实现负载均衡
 - 7.3.4 流水线和流
 - 7.3.5 异步流水线
 - 7.4 反模式
 - 7.4.1 线程饥饿
 - 7.4.2 阻塞集合无穷等待
 - 7.4.3 忘记getconsumingenumerable () 方法
 - 7.4.4 采用其他生产者 / 消费者集合
 - 7.5 设计说明
 - 7.6 相关模式

<<设计模式>>

7.7 练习

7.8 扩展阅读

附录a改写面向对象模式

附录b调试和分析并行应用程序

附录c技术概览

术语表

参考文献

<<设计模式>>

章节摘录

版权页：插图：CPU计量器能够说明问题。

例如，一个内核的使用率为100%，而其他内核都是空闲的。

或者应用程序是计算密集型的(即要占用大量CPU资源)，但你只使用了多核系统的一部分计算能力。

如何解决这种问题呢？

简而言之，答案就是并行编程。

像所有程序员一样，你可能熟谙编写顺序代码之道，但你会发现现在它不再满足你的性能要求了。

要想有效地使用系统的CPU资源，必须把应用程序分割成块，使各块在同一时间运行。

这说起来容易做起来难。

并行编程被誉为专家领域和敏感地带，很难重现软件缺陷。

每个程序员似乎都有一些关于并行编程的趣事，可能由于某个神秘的错误，程序并没有像预期那样运行。

在编写并行程序时，这些故事应该能帮助你正视面临的问题和困难。

幸运的是，已经有人提供了一些帮助。

· .NET Framework 4引入了一种新的编程模型，大大简化了并行工作。

后台是复杂算法的支持库，在多核架构中动态分配计算。

此外，Visual Studio 2010还包含调试和分析工具，以支持新的并行编程模型。

<<设计模式>>

编辑推荐

《设计模式:NET并行编程》内容丰富，几乎涵盖了并行编程的各个方面。

一方面，本书既有对并行计算理论的基础原理及架构的阐述，也有对动态任务并行机制以及流水线技术的深入探讨，更重要的是本书每一章都有丰富的实例及示例代码，这有助于读者深入了解其原理与应用。

另一方面，本书并非只是一本纯理论的书籍，因此真正善于学习的读者，应该适当地完成一些“课后练习”。

对本书示例进行自我学习和探索，我相信您能够从本书获得宝贵经验。

《设计模式:NET并行编程》适用于在 .NET Framework 上编写托管代码的程序员，包括在 Visual C#、Visual Basic 以及 Visual F# 上编写代码的程序员。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>