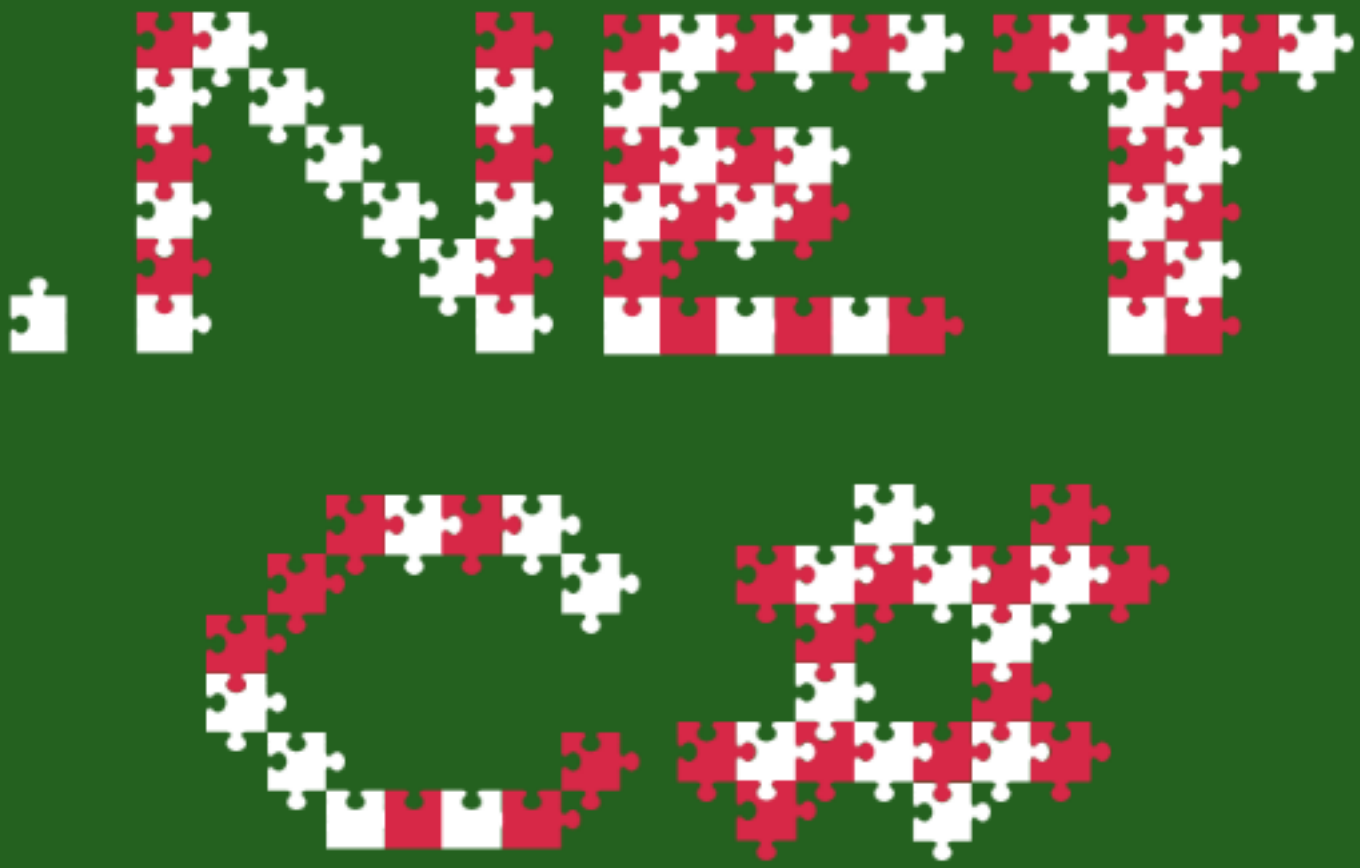
由 Parallel.For 看多執行緒程式設計

平行程式設計系列



Vulcan Lee 著

由 Parallel.For 看多執行緒程式設計

.NET / C# 平行程式設計系列

Vulcan Lee

這本書的網址是 http://leanpub.com/NET-CSharp-Parallel-Programming

此版本發布於 2021-02-27

Contents

關於本書

某日在網路上爬文，看到了臉書社團這個網址[1] 中有許多人正在熱烈討論關於如何使用 Parallel.For 這個方法來做到平行執行一萬次 Thread.Sleep(5000) 需求。這是一個很有趣的題目，看到大家在這個串列中盡情抒發自己的意見，表達各種看法，但是卻沒有人可以正確地說明如何做到提問者想要解決的問題，計算機科學領域沒有這麼模稜兩可與複雜，只要有著正確的觀念與技巧，都可以解釋當時發生了什麼問題，以及使用正確的方法來解決這些問題，然而，大家似乎無法透過這些討論內容來得到任何技能與知識提升的效果。

很早以前就想要寫本關於 .NET / C# 平行程式設計方面的書籍，不過這方面的內容相當的多與繁雜，之前已經針對這些內容整理與設計出四個階段的教育訓練課程與超過上百個講解與練習範例專案，因此，就逐一根據這個討論串列中各個人所提出的講法，設計不同的範例程式碼，而我只是想單純做個簡單的學術研究，並且將這些範例程式碼設計彙整成為一份教學課程簡報。

[1] https://bit.ly/3mNMSNJ

原本只是想針對 Parallel.For 這個題目設計一個簡單的快閃課程，有機會與大家一起來分享這些知識，但是在設計教學簡報的過程中，發現到有許多基礎的知識要能夠交代，才能夠順利解決某些問題，導致簡報內容越來越多，索性把這些過程與內容寫成一本書吧。

在此因緣際會下，開始了 .NET / C# 平行程式設計系列的第一本書 '由 Parallel.For 看多執行緒程式設計，在這本書裡面，會提供 6 個章節，將會討論到底下主題

- 我只是想做個簡單的學術研究

  針對該討論串列內容與主要關鍵核心技術先行說明一番
- 了解 Parallel.For 執行順序

  首先，先來了解 Parallel.For 是什麼，是要設計解決什麼問題
- Parallel.For 平行迴圈程式設計的指定平行工作數目上限特性

  理解平行度，也就是在進行平行資料處理，原則上會把這些資料集和切割成不同區塊，交由不同的執行緒來執行緒
- 使用 Parallel.For 做到一萬次迴圈，實際需要花多少時間

  將問題提問者提出的測試程式碼跑一次，就會知道提問者想要解決什麼問題了
- 重現造成記憶體不足情況

  當提到記憶體不足，究竟說的是哪一個？Memory Overflow / Stack Overflow ，而造成的原因是什麼？
- Parallel.For 同時執行 10000 次迴圈，觀察用了多少執行緒

  雖說 Parallel.For 可以提供資料平行處理能力，但是，究竟產生了多少執行緒來做到平行執行呢？
- Parallel.For 同時執行 200 次要花多久時間與用到多少執行緒

  Parallel.For 平行迴圈一萬次僅用到 235 執行緒，那麼，若僅平行迴圈 200 次，會不會5秒就執行完畢
- 透過執行緒集區 (預設參數) 取得過多執行緒的使用情況

  了解執行緒集區的預設執行方式

- 透過執行緒集區 (修正參數) 取得過多執行緒的使用情況

  設定執行緒集區預設準備 500 個執行緒，看看執行結果
- 使用 new Thread 來自行產生 10000 個執行緒來使用

  大家都說執行使用執行緒，效能會很好，那麼就來產生 10000 個執行緒來看看表現
- 了解同時使用 10000 個執行緒，這些執行緒開始執行時候，會造成多少延遲時間

  因為直接產生 10000 個執行緒效能表現不佳，了解問題在哪裡
- Parallel.For 透過執行緒集區準備10000個執行緒的結果

  設定執行緒集區預設準備 10000 個執行緒，執行 Parallel.For 一萬次，看看執行結果
- Parallel.For 透過執行緒集區準備不同執行緒數量的32位元執行結果

  在 32 位元下，若產生過多執行緒，會有記憶體不足問題產生
- 聽說 Task 很厲害，那就使用 10000 個 Tasks

  絕大部分的人都對 Task 與 Thread 有所誤解，在此建 10000 個 Task 來看看表現
- 用 Task.Factory.StartNew 建立 10000 個 Tasks

  使用每個 Task 都是自己生成的執行緒，不透過預設工作排程器來取得執行緒，看看表現
- 使用 Task.Delay 來取代 Thread.Sleep 的封鎖 Block 等待

  使用非封鎖的等待方式來體驗好的設計方法有什麼不同
- Parallel.For 改成使用 Task.Delay

  最後，了解如何做到資料平行 10000 次處理，最終真的僅需要 5 秒就可以執行完畢的方法

誰適合閱讀這本書

- 對於任何想要學習 .NET / C# 平行程式設計這個技術的開發者
- 需要具有 .NET / C# 開發經驗者，畢竟這本書中的例子是使用 C# 解說
- 最好能夠具備基本的計算機概念與知識

練習專案原始碼

本電子書中的練習專案原始碼，可以從 https://github.com/vulcanlee/Multi-Thread-Parallel-For-Sample[2] 取得

類型	專案名稱	專案說明
範例體驗	PF01	同時執行 10000 次
範例體驗	PF02	同時執行 10000 次，觀察用了多少執行緒?
範例體驗	PF03	那就產生 10000 個執行緒來同時執行
範例體驗	PF04	了解 10000 執行緒，執行緒開始執行會延遲多少時間
範例體驗	PF05	聽說 Task 很厲害，那就使用 10000 Tasks(Task.Run)
範例體驗	PF06	用 Task.Factory.StartNew 建立 10000 Tasks
範例體驗	PF07	使用 Task.Delay 來取代封鎖 Block 的 Thread.Sleep
範例體驗	PF08	Parallel.For 改成使用 Task.Delay
範例體驗	PF09	修正 Parallel.For 改成使用 Task.Delay 造成的錯誤
範例體驗	PF10	Parallel.For 平行迴圈程式設計的同步特性
範例體驗	PF11	同步迴圈處理特性
範例體驗	PF12	Parallel.For 平行迴圈程式設計的平行度預設特性
範例體驗	PF13	Parallel.For 平行迴圈程式設計的指定平行工作數目上限特性
範例體驗	PF14	Parallel.For 了解執行緒集區的運作方式
範例體驗	PF15	過多執行緒所造成記憶體不足的情況
範例體驗	PF16	透過執行緒集區 (預設參數) 取得過多執行緒的使用情況
範例體驗	PF17	透過執行緒集區 (修正參數) 取得過多執行緒的使用情況
範例體驗	PF18	Parallel.For 透過執行緒集區準備10000個執行緒的結果

[2] https://github.com/vulcanlee/Multi-Thread-Parallel-For-Sample

版權頁

由 Parallel.For 看多執行緒程式設計 .NET / C# 平行程式設計系列

檔案格式：EPUB3、PDF、MOBI

日期： 2021.01

作者： Vulcan Lee 李進興