

Zuerst gab es nur Hardware: ein paar Röhren und Drähte, mit denen zusammen sich nicht allzu viel anfangen ließ. Dann kam die Software dazu und erweckte die Hardware förmlich zum Leben – mit verblüffenden Ergebnissen. Mein erstes Computerprogramm, das ich auf dem einzigen Computer unserer Schule ausführte, musste noch in einer Warteschlange auf die Abarbeitung der schulspezifischen Programme zur Stundenplanerstellung und Notenverteilung warten, bis es selbst an der Reihe war. Doch schon im gleichen Jahr kamen die ersten PCs auf den Markt, was bezahlbare Computerleistung – die zuvor den Universitäten, Banken und dem Staat vorbehalten war – plötzlich in die Reichweite von Privatpersonen und -unternehmen rückte. Eine ganz neue Welt, ja sogar ungeahnte berufliche Perspektiven erschlossen sich mir mit jeder neuen wunderbaren Codezeile, die ohne Wartezeit verarbeitet wurde. Kaum war das Programm geschrieben, konnte ich schon das Ergebnis bestaunen. Bald folgte das nächste Programm, dann noch eins und noch eins und so weiter.

In der Schule lernen wir, dass mathematische Probleme linear gelöst werden. Kein Wunder also, dass das sequenzielle »Tue erst dies, dann das«-Konzept von den meisten Programmierern weltweit bevorzugt wird. Doch unsere Denkstrukturen passen nicht mehr zu den modernen Computer-Architekturen. Es werden daher neue Ansätze benötigt, weg von der linearen, sequenziellen Programmierung und hin zu der parallelen Verarbeitung, sowie neue Tools, mit denen Programmierer weiterhin schnelle, sichere, zuverlässige und direkt einsetzbare Software schreiben können – womit wir beim Thema des hier vorliegenden Werks zur parallelen Programmierung mit Microsoft Visual Studio 2010 wären.

Donis Marshall hat sein Fachwissen in Buchform gebracht und dabei fundiertes Grundlagenwissen mit praxisbezogenen Erfahrungswerten kombiniert, um auf anschauliche Weise zu zeigen, was im Bereich der parallelen Programmierung alles möglich ist. Don kann auf langjährige Berufspraxis in vielen Industriezweigen zurückblicken und hat bereits sechs Programmierbücher sowie Dutzende von Programmier tutorien verfasst, sodass es ihm spielend gelingt, Entwicklern, die noch Neulinge im Bereich der parallelen Programmierung sind, das entsprechende Grundwissen zu vermitteln und sie an die damit verbundenen Konzepte heranzuführen. Der Schritt-für-Schritt-Ansatz zusammen mit Donis' Art der Informationsvermittlung machen das Buch zu einem wertvollen Begleiter für alle Leser, die nach und nach immer tiefer in die Materie einsteigen.

Die Welt der parallelen Programmierung steht heute allen Entwicklern offen, die das Leistungspotenzial der Architekturen moderner Computer (jeglicher Form) besser ausnutzen möchten. Microsoft unterstützt sie dabei mit .NET Framework 4 und bleibt damit seiner Tradition treu, Entwicklern und Systemingenieuren stets die neuesten Funktionen und Technologien zur Verfügung zu stellen. Sie müssen sich nur informieren, wie und wo sie welche Tools am besten einsetzen. Das vorliegende Buch ist dabei eine große Hilfe und wird dazu beitragen, dass Programmierer auf der ganzen Welt die parallele Programmierung in ihr Repertoire aufnehmen können.

Tracy Monteith

Einleitung

In dieser Einleitung:

Für wen ist dieses Buch gedacht	12
Für wen ist dieses Buch nicht gedacht	12
Aufbau dieses Buchs	
Konventionen und Besonderheiten dieses Buchs	14
Systemanforderungen	14
Beispieldateien zu diesem Buch	15
Danksagung	15
Support	15
Bleiben Sie in Verbindung	15

Die parallele Programmierung wird die Software-Erstellung für die – mittlerweile allgegenwärtige – Multicore-Architektur revolutionieren und wurde zu Recht zur Schwerpunkttechnologie von Microsoft .NET Framework, Version 4, erklärt. In dieser Version von .NET Framework ist die Unterstützung für die parallele Programmierung in der TPL-Bibliothek (Task Parallel Library) und dem Namespace `System.Threading.Tasks` implementiert. Auch Microsoft Visual Studio 2010 wurde einer Überarbeitung unterzogen und umfasst jetzt Funktionen, die Sie bei der Erstellung und Wartung paralleler Anwendungen unterstützen. Wenn Sie also Microsoft-Entwickler sind und Ihre Anwendung in mehrere parallele Aufgaben zerlegen möchten, um sie auf getrennten Prozessorkernen auszuführen, dann sind Visual Studio 2010 und die TPL genau die Werkzeuge, die Sie jetzt benötigen.

Das vorliegende Buch führt Sie Schritt für Schritt in die Arbeit mit Visual Studio 2010 ein und unterstützt Sie bei der Erstellung paralleler Anwendungen. Es geht recht ausführlich auf die TPL und die Konzepte der parallelen Programmierung ein, ist aber dennoch als Einstiegslektüre zu verstehen. Hauptziel des Buchs ist es, Ihnen die Grundlagen zu den einzelnen Bereichen der parallelen Programmierung zu vermitteln, wie z.B. Aufgaben- und Datenparallelität, und Ihnen als Orientierungshilfe bei der Wahl und der Verwendung der Konzepte der parallelen Programmierung zur Seite zu stehen.

Darüber hinaus behandelt das Buch die Verwendung gleichzeitiger (threadsicherer) Auflistungen und die Threadsynchronisierung und zeigt Ihnen, wie Sie mithilfe von Visual Studio Ihre parallelen Anwendungen warten und debuggen. Ergänzt werden die Ausführungen durch zahlreiche Schritt-für-Schritt-Beispiele und herunterladbare Musterprojekte, die Sie im Selbststudium durcharbeiten können.

Für wen ist dieses Buch gedacht

Dieses Buch soll Entwicklern, die mit Microsoft Visual Basic und Microsoft Visual C# arbeiten, die Konzepte der parallelen Programmierung und verwandter Technologien näherbringen. Es richtet sich in erster Linie an all diejenigen, die die Vorteile der Multicore-Architektur nutzen möchten – ein Trend, dem sich zurzeit kein Programmierer verschließen sollte. Im Idealfall verfügt der Leser bereits über Grundkenntnisse zu .NET Framework; Erfahrungen in der parallelen Programmierung hingegen sind nicht erforderlich. Aber auch für alle, die bereits den Einstieg in die parallele Programmierung geschafft haben und vornehmlich an den neuesten Funktionen der TPL interessiert sind, kann das Buch nützlich sein.

Für wen ist dieses Buch nicht gedacht

Nicht jedes Buch ist für jedermann geeignet. Meist muss man sich als Autor darüber Gedanken machen, welche Vorkenntnisse man beim Leser voraussetzt, will man nicht fortgeschrittene Leser langweilen oder Einsteiger in das Thema abschrecken.

Vorkenntnisse

Dieses Buch geht davon aus, dass Sie zumindest über Grundkenntnisse in der .NET Framework-Entwicklung und den Konzepten der objektorientierten Programmierung verfügen. Auch wenn die TPL für die meisten, wenn nicht sogar alle, .NET Framework 4-Sprachumgebungen zur Verfügung steht, sind die Beispiele in diesem Buch ausschließlich in C# geschrieben. Sie sollten sich jedoch mit nur minimalen Änderungen in Visual Basic .NET portieren lassen. Wenn Sie keine dieser beiden Sprachen kennen, sollten Sie zuerst »Richtig einsteigen: Programmieren lernen mit Visual C# 2010« (Microsoft Press, 2011) oder »Richtig einsteigen: Programmieren lernen mit Visual Basic 2010« (Microsoft Press, 2010) von Klaus Fahnstich und Rainer G. Haselier lesen.

Da einer der Schwerpunkte dieses Buchs die gleichzeitige Programmierung ist, wird davon ausgegangen, dass Sie sich mit Threads und den Konzepten der Threadsynchronisierung auskennen. Wer tiefer in die Thread-Programmierung eindringen möchte, dem kann ich das englischsprachige Buch von Jeffrey Richter, »CLR via C#« (Microsoft Press, 2010), ans Herz legen.

Aufbau dieses Buchs

Dieses Buch besteht aus sieben Kapiteln, die jeweils einem bestimmten Aspekt der parallelen Programmierung gewidmet sind:

- Kapitel 1, »Einführung in die parallele Programmierung«, stellt die grundlegenden Konzepte der parallelen Programmierung vor
- Kapitel 2, »Aufgabenparallelität«, konzentriert sich auf die Erzeugung paralleler Aufgaben aus separaten Operationen
- Kapitel 3, »Datenparallelität«, konzentriert sich auf die Erzeugung paralleler Schleifeniterationen und dem Umwandeln von sequenziellen Schleifen in parallele Aufgaben
- Kapitel 4, »PLINQ«, gibt einen Überblick über die parallele Programmierung unter Verwendung von LINQ
- Kapitel 5, »Gleichzeitige Auflistungen«, erläutert, wie Sie mit threadsicheren Auflistungen wie `ConcurrentBag` und `ConcurrentQueue` arbeiten
- Kapitel 6, »Eigene Implementierungen«, zeigt Techniken, mit deren Hilfe Sie die TPL an individuelle Erfordernisse anpassen können
- Kapitel 7, »Berichte und Debugging«, zeigt Ihnen, wie Sie parallele Anwendungen debuggen und warten und rundet das Buch ab

Der ideale Einstiegspunkt für Sie

Das Buch deckt eine große Bandbreite an Technologien und Konzepten rund um die parallele Programmierung mit .NET Framework 4 ab. Abhängig von Ihrem aktuellen Vorhaben oder Ihrem Kenntnisstand möchten Sie das Buch womöglich nicht von vorne nach hinten durchlesen, sondern sich auf bestimmte Aspekte konzentrieren. Mithilfe der nachstehenden Tabelle können Sie feststellen, wo Sie am besten mit dem Lesen beginnen und welche Kapitel für Sie am interessantesten sein könnten.

Wenn Sie	Sollten Sie
mit den allgemeinen Konzepten der parallelen Programmierung vertraut sind	mit Kapitel 2 beginnen und den Rest des Buchs lesen
die parallelen Erweiterungen von .NET Framework 3.5 kennen	Kapitel 1 lesen, um Ihre Kenntnisse der grundlegenden Konzepte auf den aktuellen Stand zu bringen Kapitel 2 und 3 überfliegen, um sich mit den Grundlagen der Aufgaben- und Datenparallelität vertraut zu machen Kapitel 3 bis 7 lesen, um sich die Welt der TPL zu erschließen
an den LINQ-Datenanbietern interessiert sind	Kapitel 4 zu PLINQ und Kapitel 7 lesen
die TPL anpassen möchten	Kapitel 6 zu den Möglichkeiten eigener Implementierungen lesen

Tabelle E.1 Leseleitfaden

Die meisten Kapitel beinhalten praktische Beispiele die es Ihnen ermöglichen, das Gelernte auszuprobieren und umzusetzen. Gleichgültig, welche Abschnitte für Sie von Interesse sind, achten Sie darauf, die Beispielprogramme herunterzuladen und auf Ihrem System zu installieren.

Konventionen und Besonderheiten dieses Buchs

Um Ihnen die Einarbeitung in den präsentierten Stoff zu erleichtern, verwendet dieses Buch folgende Konventionen:

- Jede Übung besteht aus mehreren Schritten die angeben, was Sie tun müssen, um die Übung erfolgreich nachzuvollziehen
- Die meisten Übungsergebnisse sind in einem Konsolenfenster abgebildet, sodass Sie Ihre Ergebnisse mit den zu erwartenden Ergebnissen vergleichen können
- Am Ende jeder Übung wird der vollständige Code präsentiert. Ein Großteil des Codes steht auch zum Herunterladen zur Verfügung (siehe Hinweise im Abschnitt »Beispieldateien zu diesem Buch«)
- Code-Elemente wie `System.Threading.Tasks` werden im Fließtext durch eine besondere Schrift hervorgehoben
- Jedes Kapitel schließt mit einer Zusammenfassung der wichtigsten Aussagen des Kapitels und einer Schnellreferenz, die die wichtigsten Informationen übersichtlich präsentiert

Systemanforderungen

Um die Beispiele dieses Buchs nachvollziehen zu können, benötigen Sie die folgende Hard- und Software-Ausstattung:

- Entweder Windows XP mit Service Pack 3, Windows Server 2003 mit Service Pack 2, Windows Vista mit Service Pack 2 oder später, Windows Server 2008, Windows Server 2008 R2, Windows 7 oder Windows 7 SP1
- Visual Studio 2010, egal welche Version. (Wenn Sie mit der Express-Version arbeiten, sind unter Umständen mehrere Downloads erforderlich. Der im letzten Kapitel beschriebene Concurrency Visualizer ist erst ab der Premium-Version verfügbar.)
- Ein Computer mit einer Prozessorleistung von mindestens 1,6 GHz oder schneller (2 GHz wird empfohlen) und mehreren Prozessorkernen
- Ein Arbeitsspeicher von 1 GB (32-bit) oder 2 GB (64-bit). (Zusätzlich 1 GB, wenn die Programme in einer Virtual Machine ausgeführt werden.)
- 3,5 GB Festplattenspeicher
- Ein 5400-RPM-Festplattenlaufwerk
- Ein DVD-Laufwerk (wenn Visual Studio von einer DVD installiert wird)
- Eine Internetverbindung, um den Code zu den Übungen herunterzuladen

In Abhängigkeit von Ihrer Windows-Konfiguration benötigen Sie eventuell Administrator-Rechte, um Visual Studio 2010 installieren oder konfigurieren zu können.

Bleiben Sie in Verbindung

Beispieldateien zu diesem Buch

Zu den meisten Kapiteln in diesem Buch gibt es Übungen, die es Ihnen erlauben, das neu erworbene Wissen interaktiv auszuprobieren. Die meisten Beispielprojekte stehen zum Herunterladen unter folgender Internet-Adresse zur Verfügung:

www.microsoft-press.de/support.asp?s110=555

oder

msp.oreilly.de/support/9783866455559/671

Danksagung

Mein besonderer Dank geht an: Russell Jones für seine unendliche Geduld, Ben Ryan für die erneute wunderbare Gelegenheit, für Microsoft Press zu schreiben, und Devon Musgrave für seine »Starthilfe«. Außerdem möchte ich meinen Freunden Paul, Lynn, Cynthia, Cindy danken und allen anderen, die mich unterstützt haben. Adam, Kristen und Jason – ihr seid die hellsten Sterne im Universum.

Support

Wir haben selbstverständlich alles unternommen, um die Korrektheit dieses Buchs und des Begleitmaterials zu gewährleisten. Microsoft Press bietet Kommentare und Korrekturen für seine Bücher unter:

<http://www.microsoft-press.de/support.asp>

an.

Wenn Sie Kommentare, Fragen oder Ideen zu diesem Buch haben, senden Sie uns diese bitte per E-Mail an:

mspressde@oreilly.de

Bitte beachten Sie, dass über diese Adresse kein Support für Microsoft-Produkte angeboten wird. Wenn Sie Hilfe zu Microsoft-Produkten benötigen, kontaktieren Sie bitte den Microsoft Online Support unter

<http://support.microsoft.com>

Bleiben Sie in Verbindung

Wenn Sie mit uns Kontakt halten möchten, erreichen Sie uns über Twitter unter http://twitter.com/MSPress_de.