

Urs Gleim · Tobias Schüle

Multicore-Software

**Grundlagen, Architektur und Implementierung in
C/C++, Java und C#**



dpunkt.verlag

Urs Gleim
urs.gleim@multicore-software.de

Tobias Schüle
tobias.schuele@multicore-software.de

Lektorat: Dr. Michael Barabas
Copy-Editing: Annette Schwarz, Ditzingen
Herstellung: Birgit Bäuerlein
Umschlaggestaltung: Helmut Kraus, www.exclam.de
Druck und Bindung: M.P. Media-Print Informationstechnologie GmbH, 33100 Paderborn

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie;
detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

ISBN 978-3-89864-758-8

1. Auflage 2012
Copyright © 2012 dpunkt.verlag GmbH
Ringstraße 19 B
69115 Heidelberg

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Die Verwendung der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.

Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware-Bezeichnungen sowie Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.

Alle Angaben und Programme in diesem Buch wurden mit größter Sorgfalt kontrolliert. Weder Autor noch Verlag können jedoch für Schäden haftbar gemacht werden, die in Zusammenhang mit der Verwendung dieses Buches stehen.

5 4 3 2 1 0

Vorwort

Warum dieses Buch?

Multicore-Prozessoren verändern die Softwareentwicklung. Diese Aussage ist zurzeit vielfach zu hören. Es gibt Diskussionen über neue oder wiederentdeckte Programmiermodelle bzw. -sprachen, und Hersteller von Entwicklungswerkzeugen suchen nach höheren Abstraktionsebenen für Parallelität. Auch die in der Softwareentwicklung bereits etablierten Sprachen erhalten mit jeder neuen Version zusätzliche Funktionen und Spracherweiterungen, die dem »parallelen Zeitalter« geschuldet sind. Obwohl zurzeit niemand sagen kann, welche Ansätze sich langfristig durchsetzen werden, besteht weitgehende Einigkeit darin, dass kein Weg an Parallelität vorbeiführt. Mit diesem Buch möchten wir dem Leser die Grundlagen der Entwicklung paralleler Software an die Hand geben, die es ihm ermöglichen, die richtigen Entwurfs- und Implementierungsentscheidungen zu treffen.

An wen richtet sich das Buch?

Das Buch richtet sich an Informatiker, Softwarearchitekten und -entwickler, die einen Einblick in die Programmierung von Multicore-Systemen bekommen oder bereits vorhandene Kenntnisse vertiefen möchten. Es werden keine Vorkenntnisse über parallele Programmierung oder Multicore-Systeme vorausgesetzt. Der Leser sollte jedoch mit mindestens einer objektorientierten Programmiersprache vertraut sein und über Erfahrung in der Softwareentwicklung verfügen.

Aufbau des Buches

Der erste Teil des Buches behandelt die Grundlagen und deckt die wesentlichen Aspekte paralleler Systeme ab. Wir beginnen mit einem kurzen Streifzug durch die Architektur paralleler Prozessoren und gehen anschließend auf Threads und Synchronisationsmechanismen, die klassischen Konzepte paralleler Programmierung, ein. Im Kapitel über Task- und Datenparallelität beschreiben wir spezielle Programmiermodelle, die zunehmend Einzug in die gängigen Sprachen und Bibliotheken halten. Es folgt eine Betrachtung der für die parallele Programmierung

wichtigsten Datenstrukturen und deren Synchronisationsstrategien. Im Anschluss geben wir einen Überblick über gängige Entwurfsmuster für die Entwicklung paralleler Software und schließen den ersten Teil mit einem Kapitel über die Besonderheiten beim Entwurf paralleler Architekturen.

Der zweite Teil gibt einen kompakten Einstieg in die parallele Programmierung mit C/C++, Java und .NET. Im Kapitel über C/C++ beschreiben wir zunächst die grundlegenden Schnittstellen von Windows und Unix-basierten Systemen für die Programmierung mit Threads. Außerdem gehen wir auf die entsprechenden Mechanismen im aktuellen C++-Standard aus dem Jahr 2011 ein. Es folgt ein Überblick über die Spracherweiterung OpenMP und die C++-Bibliothek TBB. Sowohl Java als auch C# bieten in den neuesten Versionen neben den bekannten Threadschnittstellen viele Erweiterungen für die parallele Programmierung an, auf die wir in den nachfolgenden Kapiteln eingehen. Abschließend werfen wir einen Blick auf funktionale Sprachen, aktorbasierte Programmierung und transaktionsbasierten Speicher.

Ziel des zweiten Teils ist es, einen Überblick über die verfügbaren Mittel für die parallele Programmierung zu vermitteln. Wie konzentrieren uns auf die wesentlichen Konzepte und verzichten auf eine detaillierte Behandlung aller Funktionen. Dieses Buch ersetzt daher nicht die Dokumentation zu den vorgestellten Schnittstellen.

Danksagungen

Wir danken unseren Kollegen und Studenten für die vielen Diskussionen über die Inhalte dieses Buches. Herauszuheben sind Thomas Henties und Andreas Wilhelm für zeitaufwendiges Korrekturlesen und viele hilfreiche Kommentare. Aufseiten des dpunkt.verlags gilt unser Dank insbesondere Michael Barabas und allen Mitarbeitern, die zur Entstehung des Buches beigetragen haben. Außerdem danken wir Annette Schwarz für das Lektorat und Johannes Link für die konstruktiven Verbesserungsvorschläge.

Im Besonderen bedanken wir uns bei unseren Familien für die Unterstützung und die Geduld an den vielen langen Abenden und Wochenenden, an denen dieses Buch entstanden ist.

Urs Gleim und Tobias Schüle
München, November 2011

<http://www.multicore-software.de>