

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	1
1.1	Motivation	1
1.2	Begriffe der Parallelverarbeitung	4
1.3	Überblick über den Inhalt des Buches	6
<b>2</b>	<b>Architektur paralleler Plattformen</b>	9
2.1	Überblick über die Prozessorentwicklung	10
2.2	Parallelität innerhalb eines Prozessorkerns	14
2.3	Klassifizierung von Parallelrechnern	17
2.4	Speicherorganisation von Parallelrechnern	20
2.4.1	Rechner mit physikalisch verteiltem Speicher	21
2.4.2	Rechner mit physikalisch gemeinsamem Speicher	25
2.4.3	Reduktion der Speicherzugriffszeiten	27
2.5	Verbindungsnetzwerke	32
2.5.1	Bewertungskriterien für Netzwerke	34
2.5.2	Direkte Verbindungsnetzwerke	37
2.5.3	Einbettungen	43
2.5.4	Dynamische Verbindungsnetzwerke	46
2.6	Routing- und Switching-Strategien	54
2.6.1	Routingalgorithmen	54
2.6.2	Switching	66
2.6.3	Flusskontrollmechanismen	73
2.7	Caches und Speicherhierarchien	75
2.7.1	Charakteristika von Cache-Speichern	75
2.7.2	Cache-Kohärenz	86
2.7.3	Speicherkonsistenz	95
2.8	Parallelität auf Threadebene	101
2.8.1	Hyperthreading-Technik	102
2.8.2	Multicore-Prozessoren	103

2.8.3	Designvarianten für Multicore-Prozessoren . . . . .	105
2.8.4	Beispiel: Architektur des Intel Core i7 . . . . .	109
2.9	Beispiel: IBM Blue Gene Supercomputer . . . . .	112
<b>3</b>	<b>Parallele Programmiermodelle . . . . .</b>	<b>117</b>
3.1	Modelle paralleler Rechnersysteme . . . . .	118
3.2	Parallelisierung von Programmen . . . . .	121
3.3	Ebenen der Parallelität . . . . .	124
3.3.1	Parallelität auf Instruktionsebene . . . . .	124
3.3.2	Datenparallelität . . . . .	126
3.3.3	Parallelität in Schleifen . . . . .	128
3.3.4	Funktionsparallelität . . . . .	131
3.4	Explizite und implizite Darstellung der Parallelität . . . . .	133
3.5	Strukturierung paralleler Programme . . . . .	135
3.6	SIMD-Verarbeitung . . . . .	139
3.6.1	Verarbeitung von Vektoroperationen . . . . .	139
3.6.2	SIMD-Instruktionen . . . . .	141
3.7	Datenverteilungen für Felder . . . . .	143
3.8	Informationsaustausch . . . . .	148
3.8.1	Gemeinsame Variablen . . . . .	148
3.8.2	Kommunikationsoperationen . . . . .	151
3.8.3	Parallele Matrix-Vektor-Multiplikation . . . . .	158
<b>4</b>	<b>Laufzeitanalyse paralleler Programme . . . . .</b>	<b>165</b>
4.1	Leistungsbewertung von Rechnersystemen . . . . .	166
4.1.1	Bewertung der CPU-Leistung . . . . .	166
4.1.2	MIPS und MFLOPS . . . . .	168
4.1.3	Leistung von Prozessoren mit Cachespeichern . . . . .	170
4.1.4	Benchmarkprogramme . . . . .	172
4.2	Parallele Leistungsmaße . . . . .	176
4.3	Modellierung von Laufzeiten . . . . .	181
4.3.1	Realisierung von Kommunikationsoperationen . . . . .	182
4.3.2	Kommunikationsoperationen auf dem Hyperwürfel . . . . .	189
4.3.3	Kommunikationsoperationen auf einem Baum . . . . .	199
4.4	Analyse von Laufzeitformeln . . . . .	202
4.4.1	Paralleles Skalarprodukt . . . . .	203
4.4.2	Parallele Matrix-Vektor-Multiplikation . . . . .	205
4.5	Parallele Berechnungsmodelle . . . . .	208
4.5.1	PRAM-Modelle . . . . .	208
4.5.2	BSP-Modell . . . . .	210
4.5.3	LogP-Modell . . . . .	213

<b>5</b>	<b>Message-Passing-Programmierung</b> . . . . .	217
5.1	Einführung in MPI . . . . .	218
5.1.1	Einzeltransferoperationen . . . . .	220
5.1.2	Globale Kommunikationsoperationen . . . . .	234
5.1.3	Auftreten von Deadlocks . . . . .	249
5.1.4	Prozessgruppen und Kommunikatoren . . . . .	252
5.1.5	Prozesstopologien . . . . .	258
5.1.6	Zeitmessung und Abbruch der Ausführung . . . . .	263
5.2	Einführung in MPI-2 . . . . .	264
5.2.1	Prozesserzeugung und -verwaltung . . . . .	264
5.2.2	Einseitige Kommunikation . . . . .	267
<b>6</b>	<b>Thread-Programmierung</b> . . . . .	279
6.1	Einführung in die Programmierung mit Threads . . . . .	280
6.2	Programmiermodell und Grundlagen für Pthreads . . . . .	286
6.2.1	Erzeugung und Verwaltung von Pthreads . . . . .	289
6.2.2	Koordination von Threads . . . . .	292
6.2.3	Implementierung eines Taskpools . . . . .	306
6.2.4	Parallelität durch Pipelining . . . . .	310
6.2.5	Realisierung eines Client-Server-Modells . . . . .	315
6.2.6	Steuerung und Abbruch von Threads . . . . .	320
6.2.7	Thread-Scheduling . . . . .	328
6.2.8	Prioritätsinversion . . . . .	333
6.2.9	Thread-spezifische Daten . . . . .	336
6.3	Java-Threads . . . . .	337
6.3.1	Erzeugung von Threads in Java . . . . .	337
6.3.2	Synchronisation von Java-Threads . . . . .	342
6.3.3	Signalmechanismus in Java . . . . .	347
6.3.4	Erweiterte Java-Synchronisationsmuster . . . . .	351
6.3.5	Thread-Scheduling in Java . . . . .	354
6.4	OpenMP . . . . .	357
6.4.1	Steuerung der parallelen Abarbeitung . . . . .	358
6.4.2	Parallele Schleife . . . . .	361
6.4.3	Nichtiterative parallele Bereiche . . . . .	365
6.4.4	Koordination von Threads . . . . .	368
6.5	Unified Parallel C . . . . .	374
6.5.1	UPC Programmiermodell und Benutzung . . . . .	375
6.5.2	Gemeinsame Felder . . . . .	377
6.5.3	Speicherkonsistenzmodelle von UPC . . . . .	378
6.5.4	Zeiger und Felder in UPC . . . . .	380
6.5.5	Parallele Schleifen in UPC . . . . .	382
6.5.6	UPC Synchronisation . . . . .	384

<b>7</b>	<b>GPU-Programmierung</b> . . . . .	387
7.1	Überblick über die Architektur von GPUs . . . . .	387
7.2	Einführung in die CUDA-Programmierung . . . . .	395
7.3	CUDA-Synchronisation und gemeinsamer Speicher . . . . .	401
7.4	CUDA Thread Scheduling . . . . .	407
7.5	Effizienter Speicherzugriff und Tiling-Techniken . . . . .	408
7.6	Einführung in OpenCL . . . . .	414
<b>8</b>	<b>Lösung linearer Gleichungssysteme</b> . . . . .	417
8.1	Gauß-Elimination . . . . .	418
8.1.1	Beschreibung der Methode . . . . .	418
8.1.2	Parallele zeitenzyklische Implementierung . . . . .	422
8.1.3	Parallele gesamtzyklische Implementierung . . . . .	426
8.1.4	Laufzeitanalyse der gesamtzyklischen Implementierung . . . . .	432
8.2	Direkte Verfahren für Gleichungssysteme mit Bandstruktur . . . . .	437
8.2.1	Diskretisierung der Poisson-Gleichung . . . . .	438
8.2.2	Lösung von Tridiagonalsystemen . . . . .	444
8.2.3	Verallgemeinerung auf beliebige Bandmatrizen . . . . .	456
8.2.4	Anwendung auf die Poisson-Gleichung . . . . .	459
8.3	Klassische Iterationsverfahren . . . . .	461
8.3.1	Beschreibung iterativer Verfahren . . . . .	462
8.3.2	Parallele Realisierung des Jacobi-Verfahrens . . . . .	466
8.3.3	Parallele Realisierung des Gauß-Seidel-Verfahrens . . . . .	468
8.3.4	Rot-Schwarz-Anordnung . . . . .	474
8.4	Cholesky-Faktorisierung für dünnbesetzte Matrizen . . . . .	480
8.4.1	Sequentieller Algorithmus . . . . .	481
8.4.2	Abspeicherungsschemata für dünnbesetzte Matrizen . . . . .	487
8.4.3	Implementierung für gemeinsamen Adressraum . . . . .	488
8.5	Methode der konjugierten Gradienten . . . . .	497
8.5.1	Beschreibung der Methode . . . . .	498
8.5.2	Parallelisierung des CG-Verfahrens . . . . .	501
	<b>Literatur</b> . . . . .	505
	<b>Sachverzeichnis</b> . . . . .	513