

Inhaltsverzeichnis

I	Grundlagen der parallelen Programmierung	1
1	Einführung	3
1.1	Paradigmenwechsel in der Softwareentwicklung	4
1.2	Anwendungsbereiche	5
1.3	Parallelität in der Hardware	5
1.3.1	Prozessorarchitektur	6
1.3.2	Multicore-Prozessoren und Multiprozessorsysteme	9
1.4	Parallelität in der Software	13
1.4.1	Prozesse und Threads	13
1.4.2	Virtualisierung	17
1.4.3	Parallelisierende Compiler	18
1.4.4	Parallele Bibliotheken	19
1.4.5	Amdahl'sches Gesetz	20
2	Threads	23
2.1	Arbeiten mit Threads	23
2.1.1	Erzeugung und Beendigung	23
2.1.2	Datenaustausch	24
2.1.3	Threadpools	27
2.2	Scheduling	28
2.2.1	Lastverteilung	29
2.2.2	Affinitäten und Prioritäten	30
2.3	Speicherzugriff	30
2.3.1	Speichermodelle	31
2.3.2	Speicherhierarchie	37
3	Synchronisation	41
3.1	Konflikte	41
3.1.1	Entstehung	42
3.1.2	Kritische Abschnitte	43

3.2	Synchronisationsmechanismen	45
3.2.1	Mutexe	46
3.2.2	Scoped Locking	50
3.2.3	Monitore	51
3.2.4	Lese-/Schreibsperrn	52
3.2.5	Semaphore	54
3.2.6	Bedingungsvariablen	56
3.2.7	Barrieren	61
3.2.8	Einmalige Ausführung	64
3.2.9	Atomare Operationen	66
3.2.10	Spinlocks	69
3.3	Fallstricke und Richtlinien	71
3.3.1	Konflikterkennung	71
3.3.2	Verklemmungen	73
3.3.3	Nichtdeterminismus	76
3.3.4	Fairness	77
3.3.5	Skalierbarkeit	78
3.3.6	Threadsicherheit und Wiedereintrittsfähigkeit	80
3.3.7	Schnittstellenentwurf	81
4	Task- und Datenparallelität	85
4.1	Taskparallelität	85
4.1.1	Erzeugung und Synchronisation von Tasks	87
4.1.2	Parallelisierung rekursiver Algorithmen	89
4.1.3	Taskgruppen	93
4.1.4	Spekulation	95
4.1.5	Implementierung eines Task-Schedulers	98
4.1.6	Programmierrichtlinien	101
4.2	Datenparallelität	102
4.2.1	Schleifen ohne Datenabhängigkeiten	103
4.2.2	Reduktionen	106
4.2.3	Präfixberechnungen	111
4.2.4	Partitionierung und Abbildung	115
5	Datenstrukturen	127
5.1	Threadsicherer Zugriff	127
5.1.1	Grobgranulare Synchronisation	129
5.1.2	Feingranulare Synchronisation	132
5.1.3	Optimistische Synchronisation	134
5.1.4	Nichtblockierende Synchronisation	137
5.1.5	Weitere Optimierungen	139

5.2	Auswahl der richtigen Datenstruktur	139
5.2.1	Kriterien	140
5.2.2	Listen	143
5.2.3	Vektoren	144
5.2.4	Assoziative Felder	145
5.2.5	Warteschlangen und Stacks	145
5.2.6	Multimengen	147
6	Entwurfsmuster	149
6.1	Zugriff auf gemeinsame Daten	151
6.1.1	Grundlegende Synchronisationsmuster	151
6.1.2	Threadlokaler Speicher	153
6.1.3	Futures	157
6.1.4	Synchronisationsproxy	159
6.1.5	Active Object	161
6.2	Zerlegung in parallel bearbeitbare Teilprobleme	165
6.2.1	Grundlegende Zerlegungsmuster	165
6.2.2	Master-Slave	167
6.2.3	Erzeuger-Verbraucher	169
6.2.4	Aktoren	171
6.2.5	Reihenfolgebewahrender Threadpool	174
6.3	Fließbandverarbeitung	181
6.3.1	Pipelines	182
6.3.2	Pipelines mit parallelen Stufen	185
6.3.3	Parallele Pipelines	186
7	Architektur paralleler Software	191
7.1	Entwurf paralleler Algorithmen	191
7.2	Entwurf paralleler Architekturen	193

II	Sprachen und Bibliotheken	203
-----------	----------------------------------	------------

8	Threads und Synchronisation in C/C++	205
8.1	POSIX-Threads	205
8.1.1	Threads	206
8.1.2	Synchronisationsmechanismen	208
8.1.3	Threadlokaler Speicher und Speicherallokation	210
8.2	Windows-Threads	211
8.2.1	Threads	211
8.2.2	Synchronisationsmechanismen	212
8.2.3	Threadlokaler Speicher und Speicherallokation	215
8.2.4	Threadpools	215

8.3	C++11	216
8.3.1	Lambda-Funktionen	216
8.3.2	Threads	217
8.3.3	Synchronisationsmechanismen	221
8.3.4	Threadlokaler Speicher und Speicherallokation	229
8.3.5	Speichermodell	229
9	OpenMP	231
9.1	Threads	233
9.1.1	Parallele Bereiche	233
9.1.2	Arbeitsteilung	234
9.1.3	Speicherzugriff	235
9.1.4	Threadlokaler Speicher	236
9.2	Synchronisationsmechanismen	237
9.2.1	Kritische Abschnitte	237
9.2.2	Mutexe	238
9.2.3	Barrieren	238
9.2.4	Einmalige Ausführung	239
9.2.5	Atomare Operationen	240
9.3	Taskparallelität	240
9.3.1	Parallelisierung rekursiver Algorithmen	241
9.3.2	Variablenzugriff	241
9.3.3	Suspendierung	243
9.4	Datenparallelität	244
9.4.1	Schleifen ohne Datenabhängigkeiten	244
9.4.2	Reduktionen	247
9.4.3	Partitionierung und Abbildung	249
9.5	Speichermodell	250
10	Threading Building Blocks	253
10.1	Synchronisationsmechanismen	255
10.1.1	Sperren	255
10.1.2	Atomare Operationen	258
10.2	Taskparallelität	259
10.2.1	Parallele Funktionsaufrufe	259
10.2.2	Taskgruppen	260
10.2.3	Task Scheduler	261
10.3	Datenparallelität	261
10.3.1	Schleifen ohne Datenabhängigkeiten	262
10.3.2	Reduktionen	267
10.3.3	Präfixberechnungen	269

10.4	Datenstrukturen	270
10.4.1	Assoziative Felder und Mengen	271
10.4.2	Warteschlangen	274
10.4.3	Vektoren	275
10.5	Algorithmen und Entwurfsmuster	276
10.5.1	Sortieren	277
10.5.2	Fließbandverarbeitung	277
10.5.3	Flussgraphen	279
10.6	Threadlokaler Speicher	282
10.7	Speicherallokation	283
10.8	Ausnahmen und Abbruch	284
11	Parallele Programmierung mit Java	287
11.1	Threads	288
11.1.1	Threaderzeugung	288
11.1.2	Threadpools	289
11.1.3	Threadlokaler Speicher	291
11.2	Synchronisationsmechanismen	292
11.2.1	Sperren	292
11.2.2	Spezielle Synchronisationsmechanismen	296
11.2.3	Atomare Operationen	298
11.3	Taskparallelität	299
11.3.1	Tasks und Futures	299
11.3.2	Starten von Tasks	300
11.3.3	Rekursive Tasks	301
11.3.4	Blockierende Operationen in Tasks	303
11.4	Datenparallelität	303
11.5	Datenstrukturen	304
11.6	Speichermodell	305
12	Parallele Programmierung mit .NET	309
12.1	Threads	309
12.1.1	Threaderzeugung	309
12.1.2	Threadpools	311
12.1.3	Threadlokaler Speicher	313
12.2	Synchronisationsmechanismen	314
12.2.1	Sperren	314
12.2.2	Spezielle Synchronisationsmechanismen	319
12.2.3	Atomare Operationen	319
12.2.4	Einmalige Ausführung	320
12.3	Taskparallelität	321
12.3.1	Tasks und Futures	322
12.3.2	Fortsetzungstasks	324

12.4	Datenparallelität	326
12.4.1	Schleifen ohne Datenabhängigkeiten	326
12.4.2	Parallele Aggregation	328
12.4.3	PLINQ	330
12.5	Datenstrukturen	331
12.6	Ausnahmen und Abbruch	332
12.7	Speichermodell	333
13	Blick über den Tellerrand	337
13.1	Funktionale Sprachen	338
13.2	Aktorbasierte Programmierung	340
13.3	Transaktionsbasierter Speicher	342
	Literaturverzeichnis	347
	Index	351