

Inhaltsverzeichnis

1 Einführung und Grundlagen	1
1.1 Historische Entwicklung der Rechensysteme	1
1.2 Technologiefortschritte.....	5
1.2.1 Leistungsexplosion und Preisverfall der Hardware	6
1.2.2 Fortschritte bei lokalen Netzen	6
1.2.3 Aufkommen von Funkverbindungen und mobilen Geräten..	9
1.2.4 Übernetzwerk Internet	11
1.3 World Wide Web (WWW)	12
1.3.1 Web 2.0	12
1.3.2 Web 3.0	15
1.3.3 Web 4.0	16
1.3.4 E-World	17
1.3.4.1 E-Business.....	17
1.3.4.2 Weitere E-Applikationen.....	19
1.4 Selbstorganisierende Systeme	20
1.4.1 On Demand Computing.....	20
1.4.2 Autonomic Computing	22
1.4.3 Organic Computing	22
1.5 Parallele versus Verteilte Verarbeitung	23
1.5.1 Parallele Verarbeitung.....	23
1.5.1.1 Nebenläufige Prozesse.....	25
1.5.1.2 Kooperierende Prozesse	25
1.5.2 Verteilte Verarbeitung	26
1.5.2.1 Beispiele für Verteilte Systeme	27

1.5.2.2 Positive Eigenschaften der verteilten Verarbeitung	27
1.5.2.3 Eigenschaften eines Verteilten Systems.....	28
2 Rechnerarchitekturen für Parallele und Verteilte Systeme.....	33
2.1 Simultaneous Multithreading.....	34
2.1.1 Instruction Level Parallelism	34
2.1.2 Thread Level Parallelismus.....	37
2.1.3 Arbeitsweise des Simultaneous Multithreading.....	38
2.2 Eng gekoppelte Multiprozessoren und Multicore-Prozessoren .	39
2.2.1 Architektur von eng gekoppelten Multiprozessoren	39
2.2.2 Cachekohärenzprotokolle	41
2.2.2.1 MESI Cachekohärenz-Protokoll	42
2.2.2.2 Verzeichnis-basierte Cachekohärenz-Protokolle.....	51
2.2.3 Kreuzschienenschalter-basierte Multiprozessoren	52
2.2.4 Mehrebenenetzwerke-basierte Multiprozessoren	53
2.2.5 Multicore-Prozessoren.....	55
2.2.5.1 Programmierung von Multicore-Architekturen.....	59
2.2.6 Multiprozessorbetriebssysteme.....	61
2.2.6.1 Master Slave Multiprocessing.....	62
2.2.6.2 Asymmetrisches Multiprocessing.....	63
2.2.6.3 Symmetrisches Multiprocessing (SMP).....	64
2.2.6.3.1 Floating Master	65
2.2.6.4 Lock-Synchronisation.....	67
2.2.6.4.1 Test and Set (TAS)	68
2.2.6.4.2 Exchange (XCHG).....	69
2.2.6.4.3 Spinlocking	69
2.2.6.4.4 Semaphore	70
2.2.6.4.5 Compare and Swap (CAS).....	71
2.2.6.5 Transactional Memory (TM)	74

2.2.6.5.1	Programmsprachliche Realisierung des TM.....	75
2.2.6.5.2	Software Transactional Memory (STM)	77
2.2.6.5.3	Hardware Transactional Memory (HTM).....	78
2.3	Lose gekoppelte Multiprozessoren und Multicomputer	79
2.3.1	Architektur von lose gekoppelten Multiprozessoren	79
2.3.2	Verteilter gemeinsamer Speicher	81
2.3.2.1	Implementierungsebenen.....	82
2.3.2.2	Speicher Konsistenzmodelle	83
2.3.2.3	Implementierung der Sequenziellen Konsistenz	86
2.3.3	Multicomputer.....	95
2.3.4	Leistungs-Effizienzmetriken.....	96
2.4	Load Balancing und High Throughput Cluster Google	97
2.4.1	Leistungsmaße und Ausstattung des Google-Clusters.....	97
2.4.2	Google Server-Aufbau und -Architektur	99
3	Programmiermodelle für parallele und verteilte Systeme	105
3.1	Client-Server-Modell	106
3.1.1	Fehlersemantik	108
3.1.2	Serverzustände	115
3.1.3	Client-Server versus Verteilt.....	118
3.2	Service-orientierte Architekturen (SOA).....	120
3.2.1	Bestandteile eines Service	121
3.2.2	Eigenschaften eines Service	122
3.2.3	Servicekomposition, -management und -überwachung.....	125
3.2.4	Enterprise Service Bus	127
3.3	Programmiermodelle für gemeinsamen Speicher	130
3.3.1	Parallelisierende Compiler.....	136
3.3.2	Unix.....	137

3.3.2.1 fork, join	137
3.3.2.2 Erzeuger-Verbraucher (Pipe)	140
3.3.2.3 Warteschlange (Queue)	142
3.3.3 Threads	143
3.3.3.1 Threads versus Prozesse	143
3.3.3.2 Implementierung von Threads	145
3.3.3.3 Pthreads	150
3.3.3.3.1 Thread Verwaltungsroutinen.....	150
3.3.3.3.2 Wechselseitiger Ausschluss.....	153
3.3.3.3.3 Bedingungsvariable.....	156
3.3.3.3.4 Erzeuger-Verbraucher (Pipe) mit Threads.....	158
3.3.4 OpenMP	160
3.3.4.1 Parallel Pragma	162
3.3.4.2 Gültigkeitsbereiche von Daten	163
3.3.4.3 Lastverteilung unter Threads.....	164
3.3.4.3.1 for Pragma	164
3.3.4.3.2 section Pragma	165
3.3.4.3.3 single Pragma	165
3.3.4.3.4 master Pragma	166
3.3.4.4 Synchronisation	166
3.3.4.4.1 Kritische Abschnitte	166
3.3.4.4.2 Sperrfunktionen	167
3.3.4.4.3 Barriersynchronisation.....	167
3.3.5 Unified Parallel C (UPC)	168
3.3.5.1 Identifier THREADS und MYTHREAD	168
3.3.5.2 Private und Shared Data.....	169
3.3.5.3 Shared Arrays.....	169
3.3.5.4 Zeiger	170
3.3.5.5 Lastverteilung unter Threads, upc_forall.....	170

3.3.5.6 Sperrfunktionen.....	171
3.3.5.7 Barrieresynchronisation	172
3.3.5.7.1 Barrieren	172
3.3.5.7.2 Split Phase Barrieren.....	172
3.3.6 Fortress	173
3.3.6.1 Datentypen.....	173
3.3.6.2 Ausdrücke und Anweisungen	174
3.3.6.3 Juxtaposition Operator	176
3.3.6.4 Objekte, Traits, Top-level-Funktionen und Komponenten	176
3.3.6.4.1 Objekte	176
3.3.6.4.2 Traits.....	177
3.3.6.4.3 Komponenten und APIs.....	178
3.3.6.5 Parallelität.....	179
3.3.6.5.1 Schleifen und sonstige Konstrukte.....	179
3.3.6.5.2 Datenverteilung	180
3.3.6.5.3 Explizite Threads.....	180
3.3.6.5.4 Atomic-Block.....	181
3.3.7 Ada.....	182
3.3.7.1 Ada-Rendezvous	182
3.3.7.2 Selektive Ada-Rendezvous	185
3.3.7.2.1 Erzeuger-Verbraucher (Pipe) mit selektivem Rendezvous.....	186
3.3.7.3 Geschützte Objekte	188
3.4 Programmiermodelle für verteilten Speicher.....	189
3.4.1 Überblick nebenläufige Modelle	190
3.4.1.1 Nachrichtenbasierte Modelle.....	190
3.4.1.2 Datenparallelität ausnutzende Modelle	192
3.4.2 Überblick kooperative Modelle.....	194

Inhaltsverzeichnis

3.4.2.1 Lokalisierung des Kooperationspartners (Broker).....	194
3.4.2.2 Datenrepräsentation auf unterschiedlichen Maschinen	196
3.4.2.3 Nachrichtenbasierte Modelle	199
3.4.2.4 Entfernte Aufruf-Modelle.....	201
3.4.3 Nebenläufige und nachrichtenbasierte Modelle	204
3.4.3.1 Message Passing Interface (MPI).....	204
3.4.3.1.1 Initialisieren und Beenden von Prozessen	206
3.4.3.1.2 Kommunikator und Rang.....	208
3.4.3.1.3 Blockierendes Senden und Empfangen	209
3.4.3.1.4 Nichtblockierendes Senden und Empfangen	212
3.4.3.1.5 Persistente Kommunikation	214
3.4.3.1.6 Broadcast.....	216
3.4.3.1.7 Weitere kollektive Kommunikationsfunktionen....	217
3.4.3.1.8 Kommunikator und Gruppenmanagement	217
3.4.3.2 Occam.....	221
3.4.3.2.1 SEQ- versus PAR-Konstrukt	221
3.4.3.2.2 ALT-Konstrukt.....	222
3.4.3.2.3 IF- WHILE- Konstrukt, SEQ- und PAR-Zählschleifen.....	223
3.4.3.2.4 Prozeduren	224
3.4.3.2.5 Konfiguration	225
3.4.3.3 Parallel Virtual Machine (PVM)	226
3.4.3.3.1 Dämon-Prozesse	226
3.4.3.3.2 Task Erzeugung und Start.....	227
3.4.3.3.3 Hinzufügen und Entfernen von Rechnern.....	228
3.4.3.3.4 Taskkommunikation	229
3.4.3.3.5 Gruppen	232
3.4.3.3.6 Barriersynchronisation und Broadcast	233
3.4.4 Kooperative und nachrichtenbasierte Modelle	233

3.4.4.1 TCP/IP-Sockets	233
3.4.4.1.1 Datagram Sockets	237
3.4.4.1.2 Anwendungsbeispiel echo-serving	241
3.4.4.1.3 Stream-Sockets	245
3.4.4.1.4 Anwendungsbeispiel rlogin	248
3.4.4.2 Java Message Service (JMS)	251
3.4.4.2.1 Message API	252
3.4.4.2.2 Producer Consumer API	255
3.4.4.2.3 Anwendungsbeispiel Erzeuger-Verbraucher- Problem (Pipe)	264
3.4.4.2.4 JMS-Provider	269
3.4.5 Kooperative Modelle mit entfernten Aufrufen	270
3.4.5.1 Ablauf von entfernten Aufrufen	270
3.4.5.2 Abbildung des entfernten Aufrufes auf Nachrichten ...	271
3.4.5.3 Stubs	272
3.4.5.4 Parameter- und Ergebnisübertragung	273
3.4.5.5 Remote Procedure Calls (ONC RPCs, DCE RPCs, DCOM)	275
3.4.5.6 Entfernte Methodenaufrufe (CORBA)	276
3.4.5.6.1 Object Management Architecture (OMA)	277
3.4.5.6.2 Object Request Broker (ORB)	279
3.4.5.6.3 CORBA Component Model (CCM)	284
3.4.5.7 Remote Method Invocation (RMI)	285
3.4.5.7.1 Package java.rmi	286
3.4.5.7.2 Package java.rmi.registry	288
3.4.5.7.3 Package java.rmi.server	290
3.4.5.7.4 Serialisieren von Objekten	292
3.4.5.7.5 RMI-Programmierung	293
3.4.5.8 Entfernte Komponentenaufrufe	297

3.4.5.8.1 .NET Plattform	297
3.4.5.8.2 .NET Framework	299
3.4.5.8.3 .NET-Remoting	301
3.4.5.8.4 .NET 3.0.....	303
3.4.5.9 Entfernte Serviceaufrufe (Web Services)	303
3.4.5.9.1 Web Service-Architektur	304
3.4.5.10 XML-RPC.....	306
4 Parallelisierung.....	313
4.1 Leistungsmaße für parallele Programme	313
4.1.1 Laufzeit	313
4.1.2 Speedup	314
4.1.3 Kosten und Overhead	315
4.1.4 Effizienz	316
4.1.5 Amdahls Gesetz	317
4.1.6 Gustafsons Gesetz	319
4.1.7 Karp-Flatt-Metrik.....	320
4.2 Parallelisierungstechniken	321
4.2.1 Inhärenter Parallelismus.....	321
4.2.2 Zerlegungsmethoden.....	322
4.2.2.1 Funktionale Zerlegung	324
4.2.2.2 Datenzerlegung.....	325
4.2.2.2.1 Master Worker-Schema.....	326
4.2.2.2.2 Berechnungsbäume	327
4.2.2.3 Funktions- und Datenzerlegung.....	328
4.2.2.3.1 Methodisches Vorgehen	328
4.2.2.3.2 Dynamische Allokation	331
4.2.3 Weitere parallele Verfahren und Algorithmen	332

5 Verteilte Algorithmen	333
5.1 Verteilt versus zentralisiert	333
5.2 Logische Ordnung von Ereignissen	336
5.2.1 Lamport-Zeit	336
5.2.2 Vektoruhren	339
5.3 Auswahlalgorithmen	342
5.3.1 Bully-Algorithmus	342
5.3.2 Ring-Algorithmus	345
5.4 Übereinstimmungsalgorithmen	346
5.4.1 Unzuverlässige Kommunikation	348
5.4.2 Byzantinische fehlerhafte Prozesse	349
6 Rechenlastverteilung	353
6.1 Statische Lastverteilung	356
6.1.1 Jobmodelle	358
6.1.1.1 Task-Präzedenz-Graphen	359
6.1.1.2 Task-Interaktionsgraphen	360
6.1.1.3 Workflows	362
6.1.2 Lösungsverfahren	364
6.2 Dynamische Lastverteilung	368
6.2.1 Zentrale Lastverteilungssysteme	376
6.2.2 Dezentrale Lastverteilungssysteme	379
6.2.2.1 Lastausgleich ohne Migration	382
6.2.2.2 Lastausgleich mit Migration	382
6.3 Grid Scheduling	389
7 Virtualisierungstechniken	395
7.1 Betriebssystemvirtualisierung	396
7.1.1 Vollvirtualisierung	398

Inhaltsverzeichnis

7.1.2	Containervirtualisierung	400
7.1.3	Paravirtualisierung.....	400
7.2	Virtuelle Maschine.....	401
7.2.1	Java Virtuelle Maschine (JVM)	401
7.2.2	Common Language Runtime (CLR)	403
7.3	Softwarevirtualisierung	403
7.3.1	Services	403
7.3.2	Anwendungen	405
7.4	Hardware-Virtualisierung.....	405
7.4.1	Prozessor.....	406
7.4.2	Hauptspeicher.....	407
7.4.3	Datenspeicher.....	408
7.4.3.1	In-Band-Virtualisierung.....	411
7.4.3.2	Out-of-Band-Virtualisierung.....	412
7.4.4	Netzwerke	412
7.4.4.1	Virtual Local Area Network.....	413
7.4.4.2	Virtual Private Network	413
8	Cluster	415
8.1	Definition Cluster	416
8.1.1	Vor- und Nachteile von Clustern	417
8.1.2	Single System Image	417
8.1.3	Aufstellungskonzepte von Clustern	419
8.2	Klassifikationen von Clustern.....	420
8.2.1	Hochverfügbarkeits-Cluster	422
8.2.2	High Performance-Cluster	427
8.2.2.1	Beowulf	428
8.2.2.2	Wolfpack.....	430

8.2.3 Cluster für hohen Datendurchsatz	430
8.2.4 Skalierbare-Cluster.....	431
8.3 Zugangs-Konzepte.....	432
9 Grid-Computing.....	435
9.1 Definition Grid	435
9.2 Unterscheidung von Grids.....	437
9.3 Grid Middleware-Systeme.....	438
9.3.1 Globus Toolkit	439
9.3.2 gLite	440
9.3.3 Unicore	440
9.4 Weitere Grid Software	444
9.4.1 GridSphere	444
9.4.2 Shibboleth.....	444
9.4.3 VOMS.....	445
9.4.4 SRB	445
9.4.5 SRM/dCache	445
9.4.6 OGSA-DAI	446
9.4.7 GAT	446
Literatur.....	447
Schlagwortverzeichnis	481