

# Inhaltsverzeichnis

<b>1</b>	<b>Einführung und Grundlagen</b>	<b>1</b>
1.1	Historische Entwicklung der Rechensysteme	1
1.2	Technologiefortschritte	4
1.2.1	Leistungsexplosion und Preisverfall der Hardware	5
1.2.2	Fortschritte bei lokalen Netzen	6
1.2.3	Aufkommen von Funkverbindungen und mobilen Geräten	8
1.2.4	Übernetzwerk Internet	10
1.3	World Wide Web (WWW)	11
1.3.1	Web 2.0	11
1.3.2	Web 3.0	16
1.3.3	Web 4.0	17
1.3.4	E-World	18
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
1.5.2	Nebenläufige Prozesse	24
1.5.3	Kooperierende Prozesse	24
1.5.4	Verteilte Verarbeitung	25
	Literatur	30
<b>2</b>	<b>Rechnerarchitekturen für Parallele und Verteilte Systeme</b>	<b>33</b>
2.1	Eng gekoppelte Multiprozessoren und Multicore-Prozessoren	34
2.1.1	Simultaneous Multithreading	35
2.1.2	Architektur von eng gekoppelten Multiprozessoren	39
2.1.3	Programmierung von Multicore-Architekturen	54
2.1.4	Multiprozessorbetriebssysteme	55
2.2	General Purpose Computation on Graphic Processing Unit (GPGPU)	70
2.3	Many-Core-Prozessoren und Tile-Prozessor-Architekturen	73
		<b>IX</b>

2.3.1	Intel Single-chip Cloud Computer (SCC)	73
2.3.2	Tile-CPU	76
2.4	Lose gekoppelte Multiprozessoren und Cluster	80
2.4.1	Architektur von lose gekoppelten Multiprozessoren	80
2.4.2	Verteilter gemeinsamer Speicher	82
2.4.3	Multicomputer	93
2.4.4	Leistungs-Effizienzmetriken	94
2.4.5	Load Balancing und High Throughput Cluster Google	96
	Literatur	101
3	<b>Client-Server-Modell und Web-Services</b>	107
3.1	Client-Server-Modell	108
3.1.1	Fehlersemantik	109
3.1.2	Serverzustände	114
3.1.3	Client-Server versus Verteilt	117
3.2	Service-orientierte Architekturen (SOA)	118
3.2.1	Bestandteile eines Dienstes (Service)	119
3.2.2	Eigenschaften eines Dienstes (Service)	120
3.2.3	Servicekomposition, -management und -überwachung	122
3.2.4	Enterprise Service Bus (ESB)	125
	Literatur	126
4	<b>Programmiermodelle für gemeinsamen Speicher</b>	129
4.1	Überblick	129
4.2	Parallelisierende Compiler	133
4.3	Unix	134
4.3.1	fork()/join()-Parallelismus	134
4.3.2	Dynamische Prozesse	136
4.3.3	Gemeinsamer Speicher	136
4.3.4	Semaphore	137
4.3.5	Erzeuger-Verbraucher (Pipe)	137
4.3.6	Pipes in Unix	138
4.3.7	Warteschlange (Queue)	139
4.4	Threads	140
4.4.1	Threads versus Prozesse	140
4.4.2	Implementierung von Threads	141
4.4.3	Pthreads	145
4.5	OpenMP	156
4.5.1	Parallel Pragma	157
4.5.2	Gültigkeitsbereiche von Daten	158
4.5.3	Lastverteilung unter Threads	158
4.5.4	Synchronisation	161

4.5.5	Barrieresynchronisation	162
4.6	Unified Parallel C (UPC)	162
4.6.1	Identifiziert THREADS und MYTHREAD	162
4.6.2	Private und Shared Data	163
4.6.3	Shared Arrays	164
4.6.4	Zeiger	164
4.6.5	Lastverteilung unter Threads, upc_forall	165
4.6.6	Sperrfunktionen	166
4.6.7	Barrieresynchronisation	166
4.7	CUDA	167
4.7.1	Host und Device	167
4.7.2	Kernels, Thread, Block, Grid	168
4.8	Ada	171
4.8.1	Ada-Rendezvous	171
4.8.2	Selektive Ada-Rendezvous	174
4.8.3	Erzeuger-Verbraucher (Pipe) mit selektivem Rendezvous	175
4.8.4	Geschützte Objekte	177
	Literatur	178
5	<b>Programmiermodelle für verteilten Speicher</b>	181
5.1	Überblick nebenläufige Modelle	181
5.1.1	Nachrichtenbasierte Modelle	182
5.1.2	Datenparallelität ausnutzende Modelle	189
5.2	Überblick kooperative Modelle	191
5.2.1	Lokalisierung des Kooperationspartners (Broker)	191
5.2.2	Datenrepräsentation auf unterschiedlichen Maschinen	192
5.2.3	Nachrichtenbasierte Modelle	194
5.2.4	Entfernte Aufruf-Modelle	196
5.3	Message Passing Interface (MPI)	199
5.3.1	Dynamische Prozesse	199
5.3.2	Schreib- und Lesefunktionen	199
5.3.3	Implementationen von MPI	200
5.3.4	MPJ	201
5.3.5	Initialisieren und Beenden von Prozessen	201
5.3.6	Kommunikator und Rang	202
5.3.7	Blockierendes Senden und Empfangen	203
5.3.8	Nichtblockierendes Senden und Empfangen	206
5.3.9	Persistente Kommunikation	208
5.3.10	Broadcast	209
5.3.11	Barrierensynchronisation	210
5.3.12	Weitere kollektive Kommunikationsfunktionen	210
5.3.13	Kommunikator und Gruppenmanagement	211

5.3.14	Prozessgruppen	212
5.4	Occam	214
5.4.1	SEQ- versus PAR-Konstrukt	215
5.4.2	Kommunikation mit ! und ?	215
5.4.3	ALT-Konstrukt mit Wächter (Guard)	216
5.4.4	IF- WHILE- Konstrukt, SEQ- und PAR-Zählschleifen	217
5.4.5	Prozeduren	218
5.4.6	Konfiguration	219
5.5	Parallel Virtual Machine (PVM)	219
5.5.1	Dämon-Prozesse	220
5.5.2	Task Erzeugung und Start	220
5.5.3	Hinzufügen und Entfernen von Rechnern	222
5.5.4	Taskkommunikation	222
5.5.5	Gruppen	225
5.5.6	Barrieresynchronisation und Broadcast	225
5.6	Google Go	226
5.6.1	Modularisierung und Import von Paketen	226
5.6.2	Deklarationen	227
5.6.3	Datentypen	232
5.6.4	Function	235
5.6.5	Interface	236
5.6.6	Map	237
5.6.7	Channel	238
5.6.8	Kontrollstrukturen	240
5.6.9	Anwendungsbeispiel Echo-Server	242
5.6.10	Ausnahmebehandlung	243
5.7	Erlang	243
5.7.1	Module und Funktionen	243
5.7.2	Variablen, Atome, Tupel, Listen	244
5.7.3	if, case, loop	245
5.7.4	Funktionale Objekte	248
5.7.5	Nebenläufigkeit und Nachrichtenaustausch	248
5.8	Scala	253
5.8.1	Semikolon	253
5.8.2	Hallo Welt!	253
5.8.3	Modularisierung mit Paketen	256
5.8.4	Import von Paketen	256
5.8.5	Typen, Variablen und Funktionen	257
5.8.6	Objektorientierung	260
5.8.7	Traits	262
5.9	TCP/IP-Sockets	262
5.9.1	Domänen und Socketadressen	263

5.9.2	Adressierungsstruktur	263
5.9.3	Umwandlungsfunktionen	264
5.9.4	Sockettypen	265
5.9.5	Datagram-Sockets	266
5.9.6	Stream-Sockets	272
5.10	Java Message Service (JMS)	279
5.10.1	Message API	279
5.10.2	Producer-Consumer API	280
5.10.3	Anwendungsbeispiel Erzeuger-Verbraucher-Problem (Pipe)	291
5.10.4	JMS-Provider	296
5.11	Kooperative Modelle mit entfernten Aufrufen	297
5.11.1	Ablauf von entfernten Aufrufen	297
5.11.2	Abbildung des entfernten Aufrufes auf Nachrichten	297
5.11.3	Remote Procedure Calls (ONC RPCs, DCE RPCs, DCOM)	301
5.11.4	Entfernte Methodenaufrufe (CORBA)	302
5.11.5	Remote Method Invocation (RMI)	309
5.11.6	Entfernte Komponentenaufrufe (.NET)	320
5.11.7	Entfernte Serviceaufrufe (Web Services)	325
	Literatur	332
<b>6</b>	<b>Parallelisierung</b>	<b>339</b>
6.1	Leistungsmaße für parallele Programme	339
6.1.1	Laufzeit	339
6.1.2	Speedup	340
6.1.3	Kosten und Overhead	341
6.1.4	Effizienz	342
6.1.5	Amdahls Gesetz	343
6.1.6	Gustafsons Gesetz	344
6.1.7	Karp-Flatt-Metrik	345
6.2	Parallelisierungstechniken	346
6.2.1	Inhärenter Parallelismus	346
6.2.2	Zerlegungsmethoden	347
6.2.3	Weitere parallele Verfahren und Algorithmen	354
	Literatur	354
<b>7</b>	<b>Verteilte Algorithmen</b>	<b>357</b>
7.1	Verteilt versus zentralisiert	357
7.2	Logische Ordnung von Ereignissen	359
7.2.1	Lamport-Zeit	359
7.2.2	Vektoruhren	362
7.3	Auswahlalgorithmen	364
7.3.1	Bully-Algorithmus	364

7.3.2	Ring-Algorithmus	367
7.4	Übereinstimmungsalgorithmen	368
7.4.1	Unzuverlässige Kommunikation	369
7.4.2	Byzantinische fehlerhafte Prozesse	371
	Literatur	372
<b>8</b>	<b>Rechenlastverteilung</b>	<b>373</b>
8.1	Statische Lastverteilung	375
8.1.1	Jobmodelle	377
8.1.2	Lösungsverfahren	381
8.2	Dynamische Lastverteilung	385
8.2.1	Zentrale Lastverteilungssysteme	391
8.2.2	Dezentrale Lastverteilungssysteme	393
8.3	Grid Scheduling	402
	Literatur	405
<b>9</b>	<b>Virtualisierungstechniken</b>	<b>411</b>
9.1	Betriebssystemvirtualisierung	412
9.1.1	Vollvirtualisierung	413
9.1.2	Containervirtualisierung	415
9.1.3	Paravirtualisierung	415
9.2	Virtuelle Maschine	416
9.2.1	Java Virtuelle Maschine (JVM)	416
9.2.2	Common Language Runtime (CLR)	417
9.3	Softwarevirtualisierung	418
9.3.1	Services	418
9.3.2	Anwendungen	419
9.4	Hardware-Virtualisierung	419
9.4.1	Prozessor	420
9.4.2	Hauptspeicher	421
9.4.3	Datenspeicher	422
9.4.4	Netzwerke	425
	Literatur	426
<b>10</b>	<b>Cluster</b>	<b>429</b>
10.1	Historische Entwicklung der Cluster	429
10.2	Definition Cluster	430
10.2.1	Vor- und Nachteile von Clustern	431
10.2.2	Single System Image	431
10.2.3	Aufstellungskonzepte von Clustern	432
10.3	Klassifikationen von Clustern	433
10.3.1	Hochverfügbarkeits-Cluster	435

10.3.2 High Performance-Cluster . . . . .	439
10.3.3 Cluster für hohen Datendurchsatz . . . . .	442
10.3.4 Skalierbare-Cluster . . . . .	442
10.4 Zugangs-Konzepte . . . . .	442
Literatur . . . . .	445
<b>11 Grid-Computing . . . . .</b>	<b>447</b>
11.1 Definition Grid . . . . .	447
11.2 Unterscheidung von Grids . . . . .	449
11.3 Grid Middleware-Systeme . . . . .	450
11.3.1 Globus Toolkit . . . . .	450
11.3.2 gLite . . . . .	451
11.3.3 Unicore . . . . .	452
11.4 Weitere Grid Software . . . . .	454
11.4.1 GridSphere . . . . .	454
11.4.2 Shibboleth . . . . .	455
11.4.3 VOMS . . . . .	455
11.4.4 SRB . . . . .	455
11.4.5 SRM/dCache . . . . .	456
11.4.6 OGSA-DAI . . . . .	456
11.4.7 GAT . . . . .	457
Literatur . . . . .	457
<b>12 Cloud-Computing . . . . .</b>	<b>459</b>
12.1 Organisation von Cloud-Systemen . . . . .	460
12.1.1 Public Cloud . . . . .	460
12.1.2 Private Cloud . . . . .	461
12.1.3 Hybrid Cloud . . . . .	461
12.2 Kategorien von Cloud-Diensten . . . . .	461
12.2.1 Softwaredienste (Software as a Service) . . . . .	462
12.2.2 Plattformdienste (Platform as a Service) . . . . .	463
12.2.3 Infrastrukturdienste (Infrastructure as a Service) . . . . .	464
12.3 Ausgewählte Cloud-Dienste und -Lösungen . . . . .	464
12.3.1 Amazon Elastic Block Store (EBS) . . . . .	467
12.3.2 Amazon Simple Storage Service (S3) . . . . .	468
12.3.3 Google App Engine . . . . .	468
12.3.4 Eucalyptus . . . . .	469
12.4 Herausforderungen beim Cloud Computing . . . . .	472
Literatur . . . . .	472
<b>Sachverzeichnis . . . . .</b>	<b>475</b>