

## Informatik-Fachberichte 226

---

Herausgeber: W. Brauer  
im Auftrag der Gesellschaft für Informatik (GI)

Friedemann Mattern

# Verteilte Basisalgorithmen



Springer-Verlag  
Berlin Heidelberg New York  
London Paris Tokyo Hong Kong

**Autor**

Friedemann Mattern  
Fachbereich Informatik, Universität Kaiserslautern  
Postfach 3049, D-6750 Kaiserslautern

CR Subject Classification (1987): C.2.4, D.1.3, F.2.2

CIP-Titelaufnahme der Deutschen Bibliothek.

Mattern, Friedemann:

Verteilte Basisalgorithmen / Friedemann Mattern. – Berlin; Heidelberg; New York; London; Paris;

Tokyo: Springer, 1989

(Informatik-Fachberichte; 226)

Zugl.: Kaiserslautern, Univ., Diss.

ISBN-13: 978-3-540-51835-8 e-ISBN-13: 978-3-642-84002-9

DOI: 10.1007/978-3-642-84002-9

NE: GT

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland vom 9. September 1965 in der Fassung vom 24. Juni 1985 zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtsgesetzes.

© Springer-Verlag Berlin Heidelberg 1989

2145/3140 – 543210 – Gedruckt auf säurefreiem Papier

# Vorwort

*Verteilte Algorithmen* sind Verfahren, die dadurch charakterisiert sind, daß mehrere autonome Prozesse gleichzeitig Teile eines gemeinsamen Problems in kooperativer Weise bearbeiten und der dabei erforderliche Informationsaustausch ausschließlich über *Nachrichten* erfolgt. Derartige Algorithmen kommen im Rahmen *verteilter Systeme* zum Einsatz, bei denen kein gemeinsamer Speicher existiert und die Übertragungs- und Bearbeitungsdauer von Nachrichten i.a. nicht vernachlässigt werden kann. Daß dabei kein Prozeß eine aktuelle konsistente Sicht des globalen Zustands besitzt, führt zusammen mit anderen typischen Eigenschaften der Verfahren, wie *Nebenläufigkeit* und *Nichtdeterminismus*, zu interessanten Problemen.

In der vorliegenden Arbeit werden einige dieser Probleme analysiert, diskutiert und zueinander in Beziehung gesetzt. Für eine Reihe von Grundproblemen, zu denen das *Election-Problem*, das *Schnapsschußproblem* und das *Terminierungsproblem* gehören, werden verschiedene Lösungsalgorithmen angegeben und miteinander verglichen. Die Bewertung der Algorithmen umfaßt analytische und empirische Untersuchungen sowie eine Diskussion der qualitativen Eigenschaften verschiedener Varianten.

Um ein tieferes Verständnis des "Prinzips der Verteiltheit" zu erreichen, werden grundsätzliche Aspekte hervorgehoben, wie etwa die Bedeutung des Zeitbegriffs in verteilten Systemen. Darüber hinaus werden einige typische Methoden und Techniken vorgestellt, die auch für die systemnahe oder anwendungsbezogene Programmierung verteilter oder hochgradig paralleler Systeme praktisch relevant sind.

Durch die Beschreibung des Umfeldes und der wichtigsten Konzepte der verteilten Programmierung, durch weiterführende Literaturangaben und einige spielerische und gleichnishafte Beispiele werden die Problemstellungen motiviert und die Ergebnisse der Arbeit in den größeren Rahmen des Distributed Computing eingeordnet.

Das vorliegende Buch basiert auf meiner am Fachbereich Informatik der Universität Kaiserslautern durchgeführten Dissertation. Wie stets bei einer Forschungsarbeit, so beruhen auch hier die Ideen und Konzepte auf bekannten Grundlagen, erforderten eine geeignete "Entwicklungs Umgebung"<sup>1</sup> und erfuhren oft auch erst durch die Reflexionen anderer ihre endgültige Ausprägung. Mein Dank gilt in diesem Sinne meinen Kollegen – insbesondere Christian Beilken, Peter Sturm und Dieter Wybraniec – für viele spontane und anregende Diskussionen, vor allem aber Herrn

---

<sup>1</sup> Für deren praktische Komponente sei an dieser Stelle Herrn Ing. grad. R. Reske gedankt – "dem Ingenör ist nichts zu schwör" und "wer forscht, hat dorsch".

Prof. Dr. J. Nehmer, dem Leiter der Arbeitsgruppe, für die langjährige vielfältige und sehr weitgehende Unterstützung und Förderung. Die vertrauensvolle Zusammenarbeit und der von ihm geschaffene Freiraum waren für das Entstehen dieser Arbeit sehr wesentlich. Dank gebührt auch Herrn Prof. Dr. Th. Härder für die gründliche Durchsicht des Manuskripts.

Viele Diplomanden und studentische Mitarbeiter halfen durch praktische und theoretische Arbeiten auf dem Gebiet der verteilten Programmierung mit, mein Verständnis für die interessanten Probleme dezentraler Systeme zu fördern. Für ihr Interesse und ihren engagierten Einsatz sei hier insbesondere Rolf Adams, Horst Mehl, Ronny Mohr, Thomas Pfundstein, Axel Schindler, Martin Schmude und Thomas Wolf gedankt; viele andere – nicht zuletzt die Hörer meiner Vorlesung "Verteilte Algorithmen" – trugen ebenfalls zum Gelingen der Arbeit bei.

Sehr gefreut haben mich lange persönliche Gespräche und auch intensive Diskussionen, oft per Electronic Mail, mit Bernadette Charron-Bost, Uli Lemberg, Monika Mevenkamp, Michel Raynal, Michael Reinfrank, Mike Spenke und Gerard Tel, denen ich viele Hinweise zu Einzelaspekten, aber auch wertvolle allgemeine Anregungen verdanke. Erst ein so freimütiger und freundschaftlicher Gedankenaustausch kann einer Arbeit einen über den rein wissenschaftlichen Aspekt hinausgehenden Sinn geben.

Kaiserslautern, im Juli 1989

Friedemann Mattern

# Inhaltsverzeichnis

<b>1 Einleitung</b> .....	1
1.1 Verteilte Systeme .....	3
1.2 Prozesse und Ereignisse .....	6
1.3 Synchronisation und Kommunikation .....	8
1.3.1 Synchronisation .....	8
1.3.2 Kommunikation .....	10
1.4 Programmiersprachen für verteilte Systeme .....	12
1.4.1 Verteilte Programmiersprachen .....	12
1.4.2 Objektorientierte Parallelität, Actor-Modell und Atommodell .....	14
1.4.3 Die Sprache CSSA .....	17
1.4.4 Ein verteiltes Beispielprogramm .....	22
1.5 Verteilte Algorithmen .....	26
1.5.1 Beobachtungen am ggT-Beispiel .....	26
1.5.2 Verteilte Kontrollalgorithmen .....	28
1.5.3 Der Echo-Algorithmus .....	30
1.6 Ausblick auf die weiteren Kapitel .....	32
<b>2 Untersuchung von Election-Algorithmen</b> .....	35
2.1 Das Election-Problem .....	36
2.1.1 Eine Übersicht bekannter Election-Algorithmen .....	41
2.1.2 Simulationsexperimente: Zufallszahlen und Versuchsmethode .....	42
2.2 Ringbasierte Election-Verfahren .....	44
2.2.1 Election auf unidirektionalen Ringen .....	44
2.2.1.1 Eine Anwendung: Das Token-Ring-Protokoll .....	45
2.2.1.2 Nachrichtenkomplexität des unidirektionalen Verfahrens .....	47
2.2.2 Election auf bidirektionalen Ringen .....	55
2.2.2.1 Ein probabilistischer Algorithmus .....	55
2.2.2.2 Deterministische Varianten .....	57
2.2.2.3 Eine experimentelle Analyse des Koeffizienten $c$ .....	58
2.2.3 Der Einfluß nicht-konstanter Nachrichtenlaufzeiten .....	64
2.3 Election-Algorithmen für andere Topologien .....	67
2.3.1 Election auf Bäumen .....	68
2.3.2 Election auf allgemeinen Netzen .....	71
2.3.2.1 Problematik der Bewertung allgemeiner Election-Algorithmen .....	72
2.3.2.2 Die allgemeinen Election-Verfahren .....	73

2.3.2.3 Repräsentative beliebige Netze .....	94
2.3.2.4 Experimentelle Untersuchung allgemeiner Election-Verfahren .....	96
<b>3 Das Schnappschußproblem .....</b>	<b>103</b>
3.1 Konsistente globale Zustände .....	105
3.2 Das Schnappschußprinzip .....	112
3.3 Zwei symmetrische Schnappschußalgorithmen .....	115
3.4 Basisalgorithmen als Bausteine .....	118
3.5 Konsistenz durch 'Einfrieren' .....	119
<b>4 Verteilte Terminierung .....</b>	<b>121</b>
4.1 Einleitung .....	121
4.1.1 Das Märchen von der verteilten Terminierung .....	123
4.2 Ein Beispiel – verteiltes Lösen von krypto-arithmetischen Rätseln .....	125
4.2.1 Verteiltes Problemlösen .....	126
4.2.2 Sequentielle und parallele Lösungsansätze .....	127
4.2.3 Backtracking und Erkennung der Terminierung .....	129
4.2.4 Weitere Aspekte .....	131
4.3 Der Terminierungsbegriff .....	135
4.3.1 Ergebnisorientierte Terminierung .....	136
4.3.2 Kommunikationsorientierte Terminierung .....	139
4.4 Eine Analyse des Terminierungsproblems .....	145
4.4.1 Einfaches Zählen genügt nicht .....	148
4.4.2 Das Problem bei synchroner Kommunikation .....	149
4.5 Charakteristika von Terminierungsalgorithmen .....	151
4.5.1 Basisprinzipien .....	151
4.5.2 Eigenschaften, Merkmale und Kriterien .....	153
4.6 Verfahren mit zwei Wellen .....	157
4.6.1 Das Doppelzählverfahren .....	159
4.6.2 Der skeptische Algorithmus .....	161
4.6.3 Election-basierte Terminierungsalgorithmen .....	163
4.7 Zeitonenverfahren .....	172
4.7.1 Eine symmetrische ringbasierte Version .....	174
4.7.2 Eine sternförmige Version mit beschränkter Zahl von Zeitonen .....	176
4.7.3 Ein effizienter Terminierungstest für synchrone Kommunikation .....	177
4.7.3.1 Der DFG-Algorithmus .....	178
4.7.3.2 Eine effiziente Variante .....	179
4.7.3.3 Eine Realisierung in CSP .....	181
4.7.3.4 Bemerkungen zu synchronen Lösungen und CSP .....	184
4.8 Die Vektormethode .....	184
4.8.1 Der Revisor .....	185
4.8.2 Das Prinzip .....	186
4.8.3 Varianten .....	191
4.9 Die Kreditmethode .....	195

4.9.1 Eine Realisierung der Kreditmethode .....	197
4.9.2 Eine Variante .....	199
4.9.3 Optimierungen und weitere Varianten .....	200
4.10 Empirische Effizienzmessungen .....	202
4.10.1 Messungen am Zahlenrätselprogramm .....	203
4.10.2 Messungen mit synthetischen Benchmarkprogrammen .....	206
<b>5 Virtuelle Zeit in verteilten Systemen .....</b>	<b>213</b>
5.1 Zeitdiagramme und Kausalitätsstruktur .....	216
5.2 Realzeit .....	219
5.3 Virtuelle Zeit .....	221
5.4 Vektorzeit .....	224
5.4.1 Die Struktur der Vektorzeit .....	230
5.4.2 Die Analogie zu Minkowski's Raumzeit .....	237
5.4.3 Anwendungsmöglichkeiten der Vektorzeit .....	243
5.5 Stichzeitpunkt und Schnappschuß .....	246
<b>6 Schlußbemerkungen .....</b>	<b>251</b>
<b>Anhang – Meßergebnisse allgemeiner Election-Algorithmen .....</b>	<b>255</b>
<b>Literatur .....</b>	<b>271</b>
<b>Sachverzeichnis .....</b>	<b>283</b>