

Dipl.-Ing. Thomas Mike Warschko,  
Karlsruhe

# **Effiziente Kommunikation in Parallelrechner- architekturen**

Reihe **10**: Informatik/  
Kommunikationstechnik Nr. **525**

# Inhaltsverzeichnis

<b>Danksagung</b>	<b>III</b>
<b>Zusammenfassung</b>	<b>IX</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Ausgangspunkt . . . . .	1
1.2 Latenzzeitreduktion . . . . .	2
1.3 Latenzzeitverbergung . . . . .	2
1.4 Ziel der Dissertation . . . . .	3
1.5 Aufbau der Arbeit . . . . .	4
<b>2 Grundlagen und Vorbetrachtungen</b>	<b>5</b>
2.1 Grundlagen . . . . .	5
2.1.1 Schichtenmodell . . . . .	5
2.1.2 Physikalischer Aufbau eines Parallelrechners . . . . .	6
2.1.3 Architekturmodelle . . . . .	7
Architekturen mit verteiltem Speicher . . . . .	8
Architekturen mit gemeinsamem Adreßraum . . . . .	8
Architekturen mit gemeinsamem Speicher . . . . .	8
2.1.4 Modelle der Parallelität . . . . .	9
2.1.5 Programmiermodelle . . . . .	10
Prinzip der Nachrichtenkopplung . . . . .	10
Prinzip der Datenparallelität . . . . .	11
Prinzip der gemeinsamen Variablen . . . . .	12
Das PRAM-Programmiermodell . . . . .	13
2.1.6 Abbildung der Programmiermodelle auf verschiedene Architekturen . . . . .	14
Emulation der Nachrichtenkopplung . . . . .	14
Emulation der Datenparallelität . . . . .	15
Emulation der gemeinsamen Variablen . . . . .	16
PRAM-Emulation . . . . .	17
2.1.7 Konsistenzmodelle in Mehrprozessorsystemen . . . . .	17
2.2 Kommunikationslatenzzeiten . . . . .	20
2.2.1 Netzwerklatenzzeiten . . . . .	20
Leitungsvermittelte Netze . . . . .	20
Paketvermittelte Netze . . . . .	21
2.2.2 Schnittstellenlatenzzeiten . . . . .	22
2.2.3 Protokollatenzzeiten . . . . .	24
2.2.4 Betriebssystemlatenzzeiten . . . . .	25
2.2.5 Bibliothekslatenzzeiten . . . . .	26
2.3 Bekannte Verfahren zur Latenzzeitreduktion und -verbergung . . . . .	29
2.3.1 Cache-Kohärenz-Protokolle . . . . .	29

2.3.2	Modelle mit schwacher Speicherkonsistenz . . . . .	30
2.3.3	Kommunikation auf Benutzerebene . . . . .	31
2.3.4	Optimierte Netzwerkprotokolle . . . . .	31
2.3.5	Asynchrone Kommunikation . . . . .	33
2.3.6	Das LogP-Modell . . . . .	34
2.3.7	Aktive Nachrichten . . . . .	34
2.3.8	Hardwarebasierte Vorladetechniken . . . . .	35
2.3.9	Softwaregesteuerte Vorlademechanismen . . . . .	36
2.3.10	Vielfädige Prozessoren . . . . .	37
2.3.11	Entkoppelte Prozessorarchitekturen . . . . .	38
2.4	Latenzzeitbehandlung in Parallelrechnerarchitekturen . . . . .	39
2.4.1	Nachrichtengekoppelte Architekturen . . . . .	39
	Latenzzeitreduktion . . . . .	39
	Latenzzeitverbergung . . . . .	40
2.4.2	Architekturen mit gemeinsamem Adreßraum . . . . .	41
2.4.3	Architekturen mit gemeinsamem Speicher . . . . .	43
	Architekturen ohne Cachekohärenz . . . . .	43
	Cachekohärente Architekturen . . . . .	44
2.4.4	Kritik an cachekohärenten Architekturen . . . . .	45
2.5	Schlußfolgerung und weitere Vorgehensweise . . . . .	46
2.5.1	Latenzzeitreduktion in nachrichtengekoppelten Systemen . . . . .	47
2.5.2	Latenzzeitverbergung in Systemen mit gemeinsamem Adreßraum . . . . .	47
2.6	Zusammenfassung und Beiträge dieser Arbeit . . . . .	48
<b>3</b>	<b>Techniken zur Latenzzeitreduktion</b> . . . . .	<b>49</b>
3.1	Ausgangssituation und Motivation . . . . .	49
3.2	Schwachstellenanalyse und Entwurfsalternativen . . . . .	49
3.2.1	Schwachstellen von LAN-Netzwerken . . . . .	50
3.2.2	Optimierte Kommunikationsnetzwerke . . . . .	53
	Skalierbarkeit . . . . .	53
	Wegfindung . . . . .	54
	Gesicherte Datenübertragung . . . . .	55
	Reihenfolgeerhaltung . . . . .	57
	Variable Paketgrößen . . . . .	57
	Minimales Netzwerkprotokoll . . . . .	58
	Synchronisation . . . . .	58
3.2.3	Netzwerkbetrieb . . . . .	59
	Verfahren zum Ausschluß von Verklemmungen . . . . .	59
	Verklemmungsfreier Betrieb von Netzwerken mit Ringtopologie . . . . .	61
	Verklemmungsfreier Betrieb von $k$ -dimensionalen $N_i$ -Kuben . . . . .	68
3.2.4	Aufwand einer Betriebssystemintegration . . . . .	71
3.2.5	Betriebssysteminteraktion . . . . .	73
3.2.6	Schwachstellen gängiger Kommunikationsprotokolle . . . . .	74
	Kommunikationshardware . . . . .	75
	Kommunikationspartner . . . . .	75
	Netzwerkumgebung . . . . .	76
3.2.7	Optimierte Kommunikationsprotokolle . . . . .	77
	Adressierung . . . . .	78
	Datenaustausch . . . . .	79
	Verbindungsaufbau bzw. -abbau . . . . .	79
	Optimierungstechniken . . . . .	79

	Fazit . . . . .	80
3.2.8	Programmierumgebungen . . . . .	81
	Typisierte Nachrichten . . . . .	81
	Übertragung sehr großer Datenmengen . . . . .	82
	Prozeßverwaltung . . . . .	84
3.2.9	Fazit . . . . .	85
3.3	Verwandte Arbeiten . . . . .	85
3.3.1	Systeme mit gemeinsamem Speicher . . . . .	85
3.3.2	Systeme mit verteiltem Speicher . . . . .	87
3.3.3	Ähnlichkeiten und Unterschiede zu ParaStation . . . . .	90
3.4	Realisierung . . . . .	90
3.4.1	Das ParaStation-Netzwerk . . . . .	90
	Prinzipieller Aufbau des Kommunikationsnetzes . . . . .	91
	Paketaufbau . . . . .	92
	Wegfindung . . . . .	93
	Funktionsweise des Netzwerkkontrollprozessors . . . . .	94
	Synchronisation . . . . .	95
	Systemintegration . . . . .	97
	Programmierschnittstelle . . . . .	98
3.4.2	Interaktion mit dem Betriebssystem . . . . .	99
3.4.3	Die ParaStation-Kommunikationsbibliothek . . . . .	100
	Die Hardware-Abstraktions-Schicht . . . . .	101
	Die ParaStation-System-Schicht (Ports) . . . . .	102
	Die Socket-Emulation . . . . .	108
3.4.4	Zusammenspiel aller Komponenten . . . . .	111
3.4.5	Fazit . . . . .	112
3.5	Ergebnisse . . . . .	113
3.5.1	Der Evaluationsprototyp ParaPC . . . . .	113
3.5.2	ParaStation-Plattformen . . . . .	115
3.5.3	Leistungsmessung auf Kommunikationsebene . . . . .	116
	Theoretische Spitzenleistung . . . . .	116
	Messung der Kommunikationslatenzzeit . . . . .	117
	Messung des Datendurchsatzes . . . . .	119
	Vergleich unterschiedlicher Meßmethoden . . . . .	121
3.5.4	Synchronisation . . . . .	122
3.5.5	Leistung der Protokollvarianten . . . . .	123
3.5.6	Leistungsmessung auf Applikationsebene . . . . .	125
	Hitzediffusion . . . . .	126
	ScaLAPACK . . . . .	128
3.6	Ausblick . . . . .	129
3.7	Zusammenfassung und Beiträge . . . . .	130
3.7.1	Zusammenfassung . . . . .	130
3.7.2	Beiträge dieser Arbeit . . . . .	131
<b>4</b>	<b>Techniken zur Latenzzeitverbergung</b> . . . . .	<b>133</b>
4.1	Ausgangssituation . . . . .	133
4.2	Das SCAP-Verfahren . . . . .	134
4.2.1	Netzwerkanforderungen . . . . .	134
4.2.2	Systemintegration der Vorladepuffer . . . . .	135
4.2.3	Vorgehensweise des Übersetzers . . . . .	136
4.2.4	Automatische Generierung der Vorladeschleife . . . . .	138

4.2.5	Grenzen des SCAP-Verfahrens . . . . .	139
4.2.6	Fazit . . . . .	139
4.3	Verwandte Arbeiten . . . . .	139
4.4	Analytische Modellierung . . . . .	141
4.4.1	Blockierende Netzwerke . . . . .	142
4.4.2	Überlappende Netzwerke . . . . .	142
4.4.3	Analyse der Systemlaufzeiten . . . . .	148
4.4.4	Effizienzbetrachtungen . . . . .	151
4.4.5	Weitere Optimierungen . . . . .	153
	Oberflächeneffekte . . . . .	153
	Reduktionsoperationen . . . . .	153
4.5	Simulation . . . . .	154
4.5.1	Die Simulationsumgebung . . . . .	154
4.5.2	Erweiterung des Befehlssatzes . . . . .	155
4.5.3	Erweiterung des Simulators . . . . .	156
4.5.4	Transformation eines Beispielprogramms . . . . .	157
4.5.5	Validation . . . . .	160
4.5.6	Vergleich mit analytischem Modell . . . . .	161
4.6	Auswahl der Testprogramme . . . . .	164
4.6.1	Charakterisierung der Testprogramme . . . . .	165
4.6.2	Klassifikation der Testprogramme . . . . .	166
4.7	Ergebnisse . . . . .	168
4.7.1	Variation der Virtualisierung . . . . .	169
	Algorithmenklasse K1a . . . . .	169
	Algorithmenklasse K1b . . . . .	172
	Algorithmenklasse K1d . . . . .	174
	Algorithmenklasse K2a . . . . .	176
	Algorithmenklasse K2c . . . . .	178
	Fazit . . . . .	180
4.7.2	Variation der Netzwerklatenzzeit . . . . .	181
4.7.3	Variation der Prozessoranzahl . . . . .	181
4.7.4	Variation der Netzwerkkapazität . . . . .	184
4.8	Ausblick . . . . .	186
4.9	Zusammenfassung und Beiträge . . . . .	189
4.9.1	Zusammenfassung . . . . .	189
4.9.2	Beiträge dieser Arbeit . . . . .	190
<b>5</b>	<b>Konklusion und Ausblick</b>	<b>191</b>
	<b>Literaturverzeichnis</b>	<b>193</b>
	<b>Veröffentlichungen</b>	<b>209</b>