

Inhaltsverzeichnis

1	Einleitung	1
2	Modellbetrachtungen, Speicher- und Rechnerarchitekturen	5
2.1	Einleitung	5
2.2	Speedup	6
2.3	Skalierbarkeit	9
2.4	Fehler in parallelen Programmen	10
2.5	Optimierung der Einzelprozessorperformance	11
2.5.1	Fließbandverarbeitung	12
2.5.2	GFLOPs und Co.	15
2.6	Speichertechnologien und Speicherhierarchien	16
2.6.1	Speichertechnologien	17
2.6.2	Speicherhierarchien	18
2.7	Parallelrechnerarchitekturen	22
2.7.1	SMP Systeme	22
2.7.2	Parallelrechner mit verteiltem Speicher	25
2.8	Einfaches Performancemodell	28
2.8.1	Beispiel 1: Summation zweier Vektoren	29
2.8.2	Beispiel 2: Bildung eines inneren Produkts	31
2.9	Erweitertes Performancemodell	36
3	Parallele Programmierung mit OpenMP	39
3.1	Einleitung	39
3.2	Eigenschaften von OpenMP	39
3.3	Adressraumverwaltung	42
3.4	Beispiele	43
3.5	Ein Team von Threads	44
3.6	Arbeitsverteilung auf Threads	45
3.6.1	Parallelisierung von Schleifen	45
3.6.2	Weitere Direktiven zur Arbeitsverteilung	46
3.6.3	Verteilung unstrukturierter Arbeit	47
3.6.4	Beispiele für die Arbeitsverteilung	48

3.7	Direktiven zur Synchronisation	49
3.8	OpenMP und Cache Kohärenz	50
3.9	Funktionen der Laufzeitbibliothek	52
3.10	Performancebetrachtungen	53
4	Message Passing für Rechner mit verteiltem Speicher	57
4.1	Einleitung	57
4.2	MPI	58
4.2.1	Ein einführendes Beispiel	59
4.2.2	Paarweise Kommunikation	63
4.2.3	Gepackte Strukturen und konstruierte Datentypen	67
4.2.4	Kollektive Kommunikationen	72
4.2.5	Communicator	76
4.2.6	Prozesserzeugung, Prozessmanagement und Threads	77
4.2.7	Einseitige Kommunikation	78
4.2.8	Ein-/Ausgabe	82
4.2.9	MPI – Versuch eines Ausblicks	83
4.3	PVM	84
4.3.1	Einleitung	84
4.3.2	Aufbau und Funktionsweise von PVM	85
4.3.3	Ein einführendes Beispiel	86
4.3.4	Beispiel eines Master-Slave Programms	87
5	Performance der Gaußelimination und das Design paralleler Programme	93
5.1	Einleitung	93
5.2	Lösungsverfahren für Gleichungssysteme	94
5.3	Vom Lehrbuch- zum Blockalgorithmus	94
5.3.1	Exkurs: Zugriff auf mehrdimensionale Felder	96
5.3.2	Performanceanalyse von Algorithmus 5.1	98
5.3.3	Blockalgorithmus: Weniger Speicherzugriffe, mehr Performance	100
5.3.4	Exkurs: BLAS	104
5.3.5	Nahezu Peak Performance mit ATLAS	105
5.4	Design paralleler Programme	107
5.4.1	Programmzerlegung an Beispielen: Meereströmungen und Gaußelimination ...	107
5.4.2	Schritt 2: Kommunikation und Abhängigkeiten	110
5.4.2.1	Färbetechniken und zusätzliche Randzellen: Techniken zur Vermeidung lokaler Abhängigkeiten	110
5.4.2.2	Beschleunigung durch strukturierte Abhängigkeiten bei Leitfähigkeitsberechnungen	112
5.4.2.3	Dynamische und asynchrone Abhängigkeiten	113
5.4.3	Designschleife zur Optimierung der Arbeitspakete	116

5.4.4	Zuweisung auf Prozessoren	116
5.4.5	Fallstudie: Wechselwirkende Teilchen	119
5.5	Paralleles Design der Gaußelimination	123
6	Speichertechniken und die Performance iterativer Löser	131
6.1	Einleitung	131
6.2	Modellproblem: Die Poisson-Gleichung	132
6.3	Färbetechniken und das Gauß-Seidel Verfahren	134
6.4	Parallelisierung des konjugierten Gradientenverfahrens	136
6.5	Der Einfluss von Speicherungstechniken auf die Effektivität der MVM	139
6.5.1	Koordinatenformat	139
6.5.2	Komprimierte Zeilenspeicherung	141
6.5.3	Geblockte komprimierte Zeilenspeicherung	144
6.5.4	Gezackte Diagonalspeicherung	145
6.5.5	Skyline Speicherung	148
6.5.6	Diagonalspeicherung	149
6.5.7	Schiefe Diagonalspeicherung	151
6.5.8	Abschließende Betrachtung zu den Speicherungstechniken	152
6.6	Vorkonditionierungsmethoden für Parallelrechner	153
6.6.1	Einführung	154
6.6.2	Umsetzung einer ILU in komprimierter Zeilenspeicherung	154
6.6.3	Performance bei komprimierter Zeilenspeicherung	157
6.6.4	Bessere Parallelisierung mit umgekehrter Cuthill-McKee Nummerierung	159
6.6.5	ILU(0) mit Färbetechnik	162
6.6.6	Weitere Methoden zur Vorkonditionierung	165
7	Grob granulare Parallelisierung mit Gebietszerlegungsmethoden	167
7.1	Einleitung	167
7.2	Die alternierende Schwarz Methode	168
7.3	Additives und Multiplikatives Schwarz Verfahren	169
7.3.1	Parallelisierung mit Färbetechniken	171
7.3.2	Additives Schwarz Verfahren zur parallelen Vorkonditionierung	171
7.3.3	CG-Verfahren als MPI Programm	173
7.4	Gebietszerlegung ohne Überlappung	176
7.4.1	Das Schur Komplement	179
7.4.2	Parallelisierte Multiplikation des Schur Komplements mit einem Vektor	179
7.4.3	Parallelisierung des Verfahrens	183
7.4.4	Paralleles CG-Verfahren für das Gesamtgebiet	185
7.5	Multilevel Gebietszerlegungen	188

8	GPUs als Parallelrechner	191
8.1	Einleitung	191
8.2	Threadgruppen, Grid und Devicespeicher	191
8.3	Ein einführendes Beispiel	194
8.4	Speicherklassen	195
8.5	Matrix Multiplikation in Blöcken	196
8.6	Streaming Multiprozessoren und Kettenfäden	199
8.7	Aufteilung der Ressourcen	200
8.8	SIMD Architektur	201
8.9	Speicheroptimierungen	203
8.10	BLAS Funktionen auf GPUs	205
8.11	Iterative Löser auf GPUs	206
8.12	Die Zukunft von GPUs für das Parallele Rechnen	209
	Verzeichnis aufgeführter WWW-Seiten	211
	Literaturverzeichnis	213
	Index	223