

Inhaltsverzeichnis

1	Einleitung	1
1.1	Strukturierung der Arbeit	2
2	Effiziente Parallelverarbeitung	5
2.1	Single-Programming	5
2.1.1	Speedup als Maß für Effizienz	5
2.1.2	Effizientes Single-Programming auf Parallelrechnern	6
2.1.3	Grenzen des Single-Programming	6
2.2	Paralleles Scheduling	7
2.2.1	Effizienz und andere Ziele	7
	Auslastung	8
	Maximierter Durchsatz	9
	Verbesserung der mittleren Terminierungszeit	10
2.2.2	Klassifikation von Jobs	11
	Jobs mit festem Prozessorbedarf	11
	Jobs mit variablem Prozessorbedarf	11
	Jobs mit skalierbarem Prozessorbedarf	12
	reskalierbare Jobs	12
2.2.3	Methoden des Scheduling	12
	Partitionierung	13
	Time-Slicing	13
	Remapping	14
3	Scheduling paralleler Applikationen	17
3.1	Stand der Technik	17
3.1.1	Jobs mit festem Prozessorbedarf und variable Partitionierung	17
3.1.2	Jobs mit festem Prozessorbedarf und Gang-Scheduling	18
3.1.3	Skalierbare Jobs und adaptive Partitionierung	18
3.1.4	Reskalierbare Jobs und Repartitionierung	19
3.1.5	Theoretische Modelle	19

	Shelf Scheduling	19
	PBI-Scheduling	20
	Dynamic Equipartitioning	23
3.1.6	Scheduling in der Praxis	24
	EASY-Loadleveler	24
	Der PC^2 Scheduler	25
	Gang-Scheduling am LLNL	28
4	Algorithmische Skelette	33
4.1	Motivation	33
4.2	Algorithmische Skelette	34
4.3	Abstraktion durch Programmklassen	35
4.4	Stand der Technik	35
4.4.1	Allgemeine Programmierskelette	36
	Cole: Algorithmic Skeletons	36
	Higher Order Functions (HOF)	36
4.4.2	Spezielskelette	38
	Algorithmische Skelette für mathematisch-technische Anwendungen	38
	\mathcal{N} -Graphen für Transputernetzwerke	38
	PCN	39
	P^3L	40
	SCL	41
4.5	Algorithmischen Skeletten im Scheduling	43
5	Scheduling mit Skeletten	45
5.1	Skelett für Divide & Conquer	46
5.1.1	Spezifikation des D&C-Skelettes	46
5.1.2	Eigenschaften des Skelettes	47
	Skalierbarkeit	47
	Dynamisches Remapping	48
5.2	Skelett für Iterative Combination	50
5.2.1	Spezifikation des <i>ic</i> -Skelettes	50
5.2.2	Eigenschaften des Skelettes	52
	Skalierbarkeit	52
	Dynamisches Remapping	53
5.3	Skelett für Farming	54
5.3.1	Spezifikation des Farming-Skelettes	54
5.3.2	Eigenschaften des Farming-Skelettes	55
	Skalierbarkeit	55
	Dynamisches Remapping	55

5.4	Optimierung von Schedules	57
5.4.1	Speedup-Prognose skelettbasierter Programme	57
	Konservative Rechenzeitschätzung	57
	Laufzeitprognose mittels Programmklassen	59
5.4.2	Einsatz vordefinierter Datentypen	59
	Exakte Programmanalyse	59
	Effizienzverbesserung von Schedules mit grobgranularer Laufzeitprognose	60
5.4.3	Optimierung durch Reskalierung	62
5.4.4	Einfluß skalierbarer Applikationen	63
5.5	Dynamisches Remapping von Prozessoren	66
5.5.1	Integration von nicht skelettbasierten Programmen	66
5.5.2	Horizontales Remapping	67
5.5.3	Vertikales Remapping	67
5.6	Das Kostenmodell für das Scheduling	71
5.7	Optimierung mittels Back-Filling	72
6	Scheduling mit PCN	75
6.1	Einführung in PCN	75
6.1.1	Programmierung in PCN	76
6.1.2	Die Basis-Mechanismen	76
6.1.3	Datentypen und Variablen	78
6.1.4	Kommunikation und Synchronisation	80
6.1.5	Prozeß-Mapping	82
6.1.6	Das Ausführungsmodell von PCN	84
6.2	Implementierung der Skelette	85
6.2.1	Das <i>d&c</i> -Skelett	85
	H-Baum Einbettung	85
	Realisierung der Einbettungen	86
6.2.2	Das <i>ic</i> -Skelett	90
	Implementierung	90
	Test&Select-Phase	91
	Combine-Phase	91
	Internes Remapping	92
6.2.3	Das Farming-Skelett	93
	Integration von Prozessoren während der Berechnung	94
	Freigabe von Prozessoren	95
6.2.4	Das Fixed-Sized Skelett	95
6.3	Wahl des Scheduling Verfahrens	98
6.4	nichtpreemtives 2-Phasen Scheduling	99
6.4.1	Initiales Scheduling	99

6.4.2	Verteilter Ablauf des Scheduling	101
6.4.3	Verteiltes dynamisches Remapping	102
6.5	Eingesetzte Optimierungsverfahren	106
6.5.1	Ausgewählte Optimierungsverfahren	106
6.5.2	Hill-Climbing	106
	Einsatz im Scheduling	107
	Vorteile:	107
	Nachteile:	107
6.5.3	Genetische Algorithmen	108
	Einsatz im Scheduling	108
	Vorteile:	109
	Nachteile:	109
6.5.4	Der Sintflut-Algorithmus	109
	Einsatz im Scheduling	110
	Vorteile:	111
	Nachteile:	111
6.5.5	Praktischer Vergleich der Verfahren	111
6.6	Ergebnisse des Schedulingverfahrens	114
6.6.1	Laufzeitvergleich	114
6.6.2	Laufzeitmaskierungen	127
6.7	Konzept einer preemptiven Erweiterung	129
6.8	Scheduling heterogener Parallelrechner	130
6.9	Erweiterung für Parallelrechnernetze	133
7	Zusammenfassung und Ausblick	135
7.1	Zusammenfassung	135
7.2	Ausblick	136
	Algorithmische Skelette:	136
	Scheduling von Parallelrechnern:	136
A		139
A.1	Preemptive Erweiterung	139
A.2	Berechnung der H-Baum Einbettung	142

Abbildungsverzeichnis

3.1	typisches Shelf Schedule	21
3.2	Preemptives, von PBI generiertes Schedule	22
3.3	FFDH Schedule	27
3.4	FFIH Schedule	28
3.5	FFDH* Schedule	29
3.6	FFIH* Schedule	30
5.1	allgemeine Beschreibung des Divide-and-Conquer Paradigmas	46
5.2	Paralleles Profil einer D&C-Anwendung	48
5.3	allgemeine Beschreibung der iterativen Vereinigung	50
5.4	Interne Restrukturierung beim <i>ic</i> -Skelett	53
5.5	konservative Schätzung	58
5.6	Iterative Verbesserung der Schätzung	58
5.7	Auswirkung eines Jobs mit schlechter Prognose bei ungünstiger Plazierung	60
5.8	Schedule mit ungünstiger Zusammensetzung	62
5.9	Optimierung durch Reskalierung	64
5.10	Auswirkung der Optimierung mittels Reskalierung	64
5.11	Remapping zwischen einer D&C Applikation und einer Fixed- sized Anwendung	69
5.12	Remapping zwischen einer D&C Applikation und einer Farming Anwendung	69
5.13	Dynamischer Start einer Farming Applikation	70
5.14	Back-Filling mit Skalierung	72
6.1	Komposition von Programmroutinen	76
6.2	parallele Komposition	77
6.3	sequentielle Komposition	78
6.4	Fallunterscheidung	78
6.5	H-Baum Einbettung einfach (a) und optimiert (b)	85
6.6	Einbettung über eine Permutation	87

6.7	Einbettung über eine lol-Struktur	87
6.8	entarteter H-Baum	88
6.9	Permutation(10,11,2,15,9,8,1,14,5,4,3,16,13,12), 4 x 4 Gitter . .	88
6.10	Permutation (3,14,1,6,11,15,2,5,10,14,9,7,12,16), 2 x 8 Gitter .	89
6.11	Bestimmung des Ringes bei verschiedenen Gitterdimensionen .	90
6.12	physikalischer und virtueller Ring	91
6.13	Informationsobjekt für den Single-Tour Algorithmus	91
6.14	Generierung des Schedules	100
6.15	Remapping von Prozessoren	102
6.16	Schedule 1	117
6.17	Schedule 1 nach Optimierung	117
6.18	Schedule 2	118
6.19	Schedule 2 nach Optimierung	118
6.20	Schedule 3	119
6.21	Schedule 3 nach Optimierung	119
6.22	Schedule 4	120
6.23	Schedule 4 nach Optimierung	120
6.24	Schedule 5	121
6.25	Schedule 5 nach Optimierung	121
6.26	Schedule 6	122
6.27	Schedule 6 nach Optimierung	122
6.28	Schedule 7	123
6.29	Schedule 7 nach Optimierung	123
6.30	Schedule 8	124
6.31	Schedule 8 nach Optimierung	124
6.32	Schedule 9	125
6.33	Schedule 9 nach Optimierung	125
6.34	Schedule 10	126
6.35	Schedule 10 nach Optimierung	126
6.36	Slotting	130
6.37	schlechte Ausnutzung	131
6.38	gute Ausnutzung	133