

Modulverantwortliche/r	Prof. Dr. Lena Oden				
	<table border="0" style="width: 100%;"> <tr> <td style="text-align: center;">Dauer des Moduls ein Semester</td> <td style="text-align: center;">ECTS 10</td> <td style="text-align: center;">Workload 300 Stunden</td> <td style="text-align: center;">Häufigkeit in jedem Sommersemester</td> </tr> </table>	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden	Häufigkeit in jedem Sommersemester
Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden	Häufigkeit in jedem Sommersemester		
Lehrveranstaltung(en)	Advanced Parallel Computing				
Detaillierter Zeitaufwand	Bearbeiten der Lektionen: 150 Stunden, Bearbeiten der Einsendearbeiten: 75 Stunden, Studientage und Prüfungsvorbereitung: 75 Stunden				
Qualifikationsziele	Nachdem die Studierenden das Modul bearbeitet haben, können sie Modellierungswerkzeuge für parallele Programme erklären, Scheduling-Algorithmen für homogene und heterogene Zielsysteme im Umfeld statischer und dynamischer Anwendungsszenarien klassifizieren, die Auslastung von Mehrkernprozessoren analysieren und optimieren, Programmiertechniken für hybride Zielarchitekturen anwenden, die Beschränkungen der Virtualisierung identifizieren, unterschiedliche Arten der Emulation von Befehlssatzarchitekturen beurteilen, die verschiedenen Varianten virtueller Maschinen klassifizieren und gegenüberstellen, selbst virtuelle Maschinen konfigurieren, geeignete Anwendungen kategorisieren und auf einer passenden virtuellen Maschine integrieren, Schwachstellen der Virtualisierung identifizieren, Optimierungsmöglichkeiten entdecken, die Virtualisierung in geeigneter Form reorganisieren und schließlich für die jeweilige Anwendung eine bestmöglich angepasste virtuelle Plattform produzieren.				
Inhalte	<p>In der Lehrveranstaltung werden zunächst Modellierungswerkzeuge für parallele Programme eingeführt. Darauf aufbauend werden statische und dynamische Scheduling-Verfahren vorgestellt, die bei hochperformanten Parallelrechnern eine automatisierte Zuordnung der Tasks zu den einzelnen Prozessoren ermöglichen. Weiterhin werden Programmiertechniken für innovative parallele Architekturen eingeführt. Hierbei wird auch ausführlich auf die Programmierung von Graphical Processing Units (GPUs) eingegangen. In der Praxis werden parallele Implementierungen vor allem zur Lösung komplexer Optimierungsprobleme benötigt. Daher werden sowohl Approximationsverfahren als auch Heuristiken für numerische und kombinatorische Problemstellungen ausführlicher behandelt und analysiert. Schließlich wird anhand von Beispielen deren Einsatz im Umfeld von Forschung und Industrie vorgestellt.</p> <p>Ergänzende Literatur:</p> <p>E. Alba, Parallel Metaheuristics, Wiley, 2005 C. Bishof et al., Parallel Computing, Architectures, Algorithms, and Applications, IOS Press, 2008 C. Blum and A. Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. ACM Comput. Surv. , 35(3):268–308, September 2003. T. G. D. Kirk and Wen-mei Hwu, Programming Massively Parallel Processors – A Hands-on Approach, Morgan Kaufman, 2010 M. Drozdowski, Scheduling for Parallel Processing, Springer-Verlag, 2009 Yu-Kwong Kwok and I. Ahmad. Static scheduling algorithms for allocating directed task graphs to multiprocessors. ACM Comput. Surv. , 31(4):406–471, December 1999.</p>				
Inhaltliche Voraussetzung	Kenntnisse der Inhalte des Moduls 63712 "Parallele Programmierung"				
Lehr- und Betreuungsformen	Lehrveranstaltungsmaterial internetgestütztes Diskussionsforum Einsendeaufgaben mit Korrektur und/oder Musterlösung Betreuung und Beratung durch Lehrende				

Anmerkung -

Verwendung des Moduls M.Sc. Data Science
M.Sc. Informatik
M.Sc. Praktische Informatik
M.Sc. Wirtschaftsinformatik

Prüfungsformen Art der Prüfungsleistung Voraussetzung

Prüfung benotete mündliche Prüfung (ca. 25 keine
Stellenwert s. PO Minuten)
der Note