

Lehrform (*teaching format*) / **SWS** (*hours per week*): 3VL + 1UE

Kreditpunkte (*credit points*): 6

Turnus (*frequency*): i.d.R. jedes SoSe

Inhaltliche Voraussetzungen: Algorithmisches Denken. Gewisse Programmierfähigkeiten in C (empfohlen wird das "Propädeutikum C/C++", falls noch keine C-Kenntnisse vorhanden sind).

Content-related prior knowledge/skills: *Prior knowledge or skills:* Fondness of algorithmic thinking; Some basic programming skills in C (attending the course "Propädeutikum C/C++" before taking this course is recommended, if you don't have any experience in C).

Sprache (*language*): Deutsch / English

Lehrende (*teaching staff*): AG Computergraphik (Prof. Dr. Gabriel Zachmann)

Studiengang (<i>degree program</i>)	Module	Semester
Informatik (Master)	IMVP, IMVP-SQ, IMVP-AI, IMVP-DMI, IMVP-VMC	ab 1.Sem.
Digital Media (Master)	DMM-MI	ab 1.Sem.
Systems Engineering I/II (Master)	M07-VT-AuR, M07-VT-ESS	ab 1./2.Sem.
Management Information Systems (Master)	(MIS-INF3)	from 2nd sem.
Informatik (Bachelor)	(nur <i>Freie Wahl</i>)	

Lernergebnisse:

Am Ende dieser Vorlesung werden Studenten

- aktive Erfahrungen bei der Entwicklung von Software und Algorithmen für massiv-parallele Architekturen gesammelt haben
- eine Anzahl von massiv-parallelen Algorithmen-Patterns kennen
- in der Lage sein, eigene massiv-parallele Algorithmen zu entwickeln
- CUDA kennen.

Learning Outcome:

At the end of this course, students will

- have gained hands-on experience in developing software and algorithms for massively parallel computing architectures;
- have learned a number of massively-parallel algorithm patterns;
- be able to develop their own massively parallel algorithms
- be proficient in CUDA.

Inhalte:

Die Ära der single-core Prozessoren ist zu Ende. Inzwischen gibt es neue, massiv-parallele Prozessoren (GPUs), die hunderte bis tausende von Threads parallel abarbeiten können. Diese entwickeln sich zur Zeit als Co-Prozessoren, die große Teile der Berechnung den (multi-core) CPUs abnehmen. Möglicherweise werden sich GPUs als neue Architektur für die Haupt-Prozessoren – gerade auch auf mobilen Geräten – etablieren, da diese mehr Compute-Power pro Energieeinheit bieten.

Die große Zahl von parallelen Cores stellt das Design von Algorithmen und Software allerdings vor neue Herausforderungen, damit diese von der großen Parallelität profitieren können. Das Hauptziel dieser Vorlesung ist es, Studentinnen in die Lage zu versetzen, Algorithmen für solch massiv-parallele Hardware zu entwerfen.

Es gibt inzwischen viele wissenschaftliche Bereiche, in denen Studentinnen das Wissen, das sie in dieser Vorlesung erwerben, gewinnbringend einsetzen können, wie z.B.:

- Computer science (e.g., visual computing, database search)
- Computational material science (e.g., molecular dynamics simulation)
- Bio-informatics (e.g., alignment, sequencing, etc.)
- Economics (e.g., simulation of financial models)
- Mathematics (e.g., solving large PDEs)
- Mechanical engineering (e.g., CFD and FEM)
- Physics (e.g., ab initio simulations)
- Logistics (e.g. simulation of traffic, assembly lines, or supply chains)

Diese Vorlesung führt Studenten in die grundlegenden und einige fortgeschrittene Methoden und Techniken der massiv-parallelen Algorithmen ein. Einige der vorgesehenen Themen sind:

- die Programmierumgebung CUDA C;
- die Speicher-Hierarchie, verschiedene Speicher-Charakteristika, die GPU Architektur
- Parallele Reduktion, coalesced memory access
- Massiv-parallele Matrix-Algorithmen
- Prefix-Sum und deren Anwendungen in der Bildverarbeitung, Textur-Filterung
- Paralleles Sortieren (odd-even, bitonic, adaptive bitonic)
- Parallele Klassifikation mittels Machine Learning
- Thrust.

Zu Beginn der Vorlesung werden Studentinnen sich anhand von kleinen und mittelgroßen Übungen und Frameworks mit der parallelen Programmier-Umgebung CUDA-C vertraut machen. Die Übungsaufgaben werden zunehmend komplexer werden.

Contents:

There are big changes afoot. The era of increased performance from faster single cores and optimized single core programs has ended. Instead, highly parallel GPU cores, initially developed for shading, can now run hundreds or thousands of threads in parallel. Consequently, they are increasingly being adopted to offload and augment conventional (albeit multi-core) CPUs. And the technology is getting better, faster, and cheaper. It will probably even become a general computing processor on mobile devices, because it offers more processing power per energy amount.

The high number of parallel cores, however, poses a great challenge for software and algorithm design that must expose massive parallelism to benefit from the new hardware architecture. The main purpose of the lecture is to teach practical algorithm design for such parallel hardware.

There are many scientific areas where the knowledge students will gain in this course can be very valuable and useful, such as:

- Computer science (e.g., visual computing, database search)
- Computational material science (e.g., molecular dynamics simulation)
- Bio-informatics (e.g., alignment, sequencing, ...)
- Economics (e.g., simulation of financial models)
- Mathematics (e.g., solving large PDEs)
- Mechanical engineering (e.g., CFD and FEM)
- Physics (e.g., ab initio simulations)
- Logistics (e.g. simulation of traffic, assembly lines, or supply chains)

This course will introduce students to basic and some advanced methods and techniques of massively-parallel algorithms, such as:

- Programming environment CUDA C
- The memory hierarchy, different memory characteristics, the GPU architecture
- Parallel reduction, coalesced memory access
- Massively parallel matrix algorithms
- Prefix sum and applications in image processing, texture filtering
- Parallel sorting (odd-even, bitonic, adaptive bitonic)
- Dynamic parallelism
- Parallel machine learning (classification)
- Thrust.

Exercises will allow students to familiarize themselves with the CUDA parallel programming model and environment. Based on skeleton programs provided by the teacher, students will implement simple massively-parallel algorithms in CUDA. (The skeleton programs are designed to help students focus on the essential parts of the exercises.) Team development (by 2 or 3) is welcome.

Hinweise (*remarks*): In der Tabelle sind nur die primären/spezifischsten Module aufgelistet, denen diese Veranstaltung zugeordnet ist.