

> Implementierung paralleler Programmiermuster auf modernen many-core Systemen in OpenCL

Projektseminar im WiSe 2014/2015

Prof. Sergei Gorlatch, Ari Rasch, Michael Haidl





Was ist unser Ziel?

Umsetzung typischer **paralleler Programmiermuster** auf modernen **many-core Systemen**, wie Beispielsweise der Nvidia Tesla K20 oder der Intel Xeon Phi, mit **OpenCL**

Es ergeben sich folgende Fragen:

- Was sind **many-core Systeme** und **OpenCL**?

Systeme bestehend aus vielen, parallel zueinander laufenden Berechnungseinheiten, welche durch koordinierte Zusammenarbeit ein Problem schneller lösen

Beispiele:

- multi-core CPUs
- GPUs

OpenCL ist eine uniforme, jedoch hardwarenahe Schnittstelle, zur Programmierung von many-core Systemen

Es ergeben sich folgende Fragen:

- Was sind **parallele Programmiermuster**?

High-Level Abstraktion der nativen Programmierschicht (z.B.
OpenCL)

Ziele:

- abstrahieren von technischen Details (wie Datentransfer und Synchronisation)
- Reduzierung der Problematik auf ein algorithmisches Level

Beispiel:

$$\text{map}[f](a_1, \dots, a_n) = (f(a_1), \dots, f(a_n))$$

Es ergeben sich folgende Fragen:

- Was sind die Herausforderungen des Projekts?

Berücksichtigung device-spezifischer Charakteristika für eine optimierte Implementierung

Wichtig, denn Devices können technisch sehr unterschiedlich aufgebaut sein:



Grün: Recheneinheiten Gelb: Kontrolllogik Orange: Speicher



- Projekt ausgelegt für Master- und Bachelor-Studenten
 - Ma. → 20 Credit Points
 - Ba. → 10 Credit Points
- Platz für maximal 9 Studenten
- Vorkenntnisse:
 - Perfekt wäre: Erfahrung in OpenCL Programmierung
 - Optimal wäre: C++ Programmierung, Besuchte Vorlesung: PS, MGPP
 - Erwartet wird: Teamfähigkeit, Motivation, Interesse für parallele Programmierung
- Vorbesprechung:
Mittwoch, der 23.07.2014 im Lichthof 6.Stock Einsteinstr. 62