



Vorlesung Parallele Programmierung

Die parallele Programmierung gewinnt immer mehr an Bedeutung, da heutzutage bereits Mobiltelefone und Laptops über mehrere Prozessorkerne verfügen. Supercomputer besitzen teilweise sogar mehrere Millionen Kerne und haben sich als ein nützliches und mittlerweile unverzichtbares Werkzeug für viele Wissenschaftsbereiche etabliert. Die dadurch möglichen Analysen und Simulationen haben es erlaubt, den wissenschaftlichen Erkenntnisgewinn in vielen Bereichen deutlich zu steigern.

Die optimale Nutzung dieser Komponenten ist allerdings keine einfache Aufgabe, weshalb Wissenschaftlerinnen und Wissenschaftler bei der Entwicklung effizienter Anwendungen vor immer neue Herausforderungen gestellt werden. Für die parallele Programmierung ist daher ein tiefgehendes Verständnis der Hard- und Softwareumgebung sowie möglicher Fehlerursachen unabdingbar.

In der Vorlesung werden die Grundlagen der parallelen Programmierung gelehrt; die Übungen dienen der praktischen Anwendung und Umsetzung der erworbenen Kenntnisse in der Programmiersprache C.

Im Rahmen der Vorlesung werden einige der wichtigsten Themengebiete betrachtet: Hard- und Softwarekonzepte (Mehrkernprozessoren, Prozesse/Threads, NUMA etc.), unterschiedliche Ansätze zur parallelen Programmierung (OpenMP, POSIX Threads, MPI) sowie Werkzeuge zur Leistungsanalyse und Fehlersuche (Skalierbarkeit, Deadlocks, Race Conditions etc.). Zusätzlich werden Gründe und Lösungsansätze für Leistungsprobleme diskutiert und alternative Ansätze für die parallele Programmierung vorgestellt. Beispiele und Probleme werden anhand realer wissenschaftlicher Anwendungen veranschaulicht.

Veranstaltungsdaten

- Vorlesung: Dienstag 15-17 Uhr
- Übungen: Mittwoch 9-11 Uhr und Donnerstag 9-11 Uhr
- Ort: Online ([BigBlueButton](#))
- Kontakt: [Michael Kuhn](#)

Lernziel

Die Teilnehmenden lernen, parallele Programme mit verschiedenen Programmieransätzen zu erstellen, zur Ausführung zu bringen und im Ablauf zu optimieren. Außerdem werden weitere Konzepte zur Parallelisierung vermittelt und in den Übungen praktisch umgesetzt.

Voraussetzungen

Notwendige Voraussetzungen für die Vorlesung:

- Praktische Kenntnis einer Programmiersprache und die Fähigkeit, einfache Programme zu erstellen

Empfohlene Vorkenntnisse:

- Kenntnis der Grundmechanismen von Betriebssystemen
- Grundkenntnisse in Rechnerarchitekturen

Zeitplan der Vorlesung (vorläufig)

- 27.10.2020: Introduction ([Folien](#))
- 03.11.2020: Performance Analysis and Optimization ([Folien](#), [Materialien](#))
- 10.11.2020: Hardware Architectures ([Folien](#))
- 17.11.2020: Parallel Programming ([Folien](#), [Materialien](#))

24.11.2020: Programming with OpenMP ([Folien](#), [Materialien](#))

01.12.2020: Betriebssystemkonzepte

08.12.2020: Programmierung mit POSIX Threads

15.12.2020: Fehlersuche

22.12.2020: Vernetzung

05.01.2021: Programmierung mit MPI

12.01.2021: Skalierbarkeit

19.01.2021: Wissenschaftliche Anwendungen

26.01.2021: Parallele Ein-/Ausgabe

02.02.2021: Zukünftige Entwicklungen

Zeitplan der Übungen (vorläufig)

26.10.2020: Einführung ([Übungsblatt 0](#), [Übungsblatt 1](#), [Materialien](#))

02.11.2020: Debugging ([Übungsblatt 2](#), [Materialien](#))

09.11.2020: Leistungsoptimierung einer seriellen Anwendung ([Übungsblatt 3](#), [Materialien](#))

16.11.2020: Parallelisierungsschema ([Übungsblatt 4](#), [Materialien](#))

23.11.2020: Parallelisierung mit OpenMP ([Übungsblatt 5](#))

07.12.2020: Parallelisierung mit POSIX Threads

04.01.2021: MPI-Einführung

11.01.2021: Parallelisierung mit MPI

25.01.2021: Leistungsanalyse

Literatur

High Performance Computing: Modern Systems and Practices (Thomas Sterling, Matthew Anderson and Maciej Brodowicz)

Parallel Programming: for Multicore and Cluster Systems (Thomas Rauber and Gudula Rüniger)

Parallel Programming: Concepts and Practice (Dr. Bertil Schmidt, Dr. Jorge Gonzalez-Dominguez, Christian Hundt and Moritz Schlarb)