



Bachelorarbeiten

Numerik auf GPUs

Grafikkarten besitzen bis zu mehreren hundert Prozessoren, einen Arbeitsspeicher von einigen GB und zeichnen sich durch schnelle Prozesskommunikation aus. Damit eignen sie sich für rechenintensive Aufgabenstellungen bei moderatem Datenumfang. Mit der Compute Unified Device Architecture (CUDA) wurde eine Programmierumgebung geschaffen, die das Potential der Graphics Processing Unit (GPU) für numerische Berechnungen erschließt. Inzwischen existiert auch eine Matlab-Schnittstelle, welche über CUDA die Durchführung von Berechnungen auf der GPU erlaubt.

Ziel der Arbeiten ist es, für ausgewählte numerische Fragestellungen numerische Berechnungen auf der GPU zu implementieren und das Potential im Vergleich zu herkömmlichen Berechnungen auf der CPU zu untersuchen.

Es können maximal zwei Studierende an dieser Aufgabe arbeiten. Die Studierende sollen sich gegenseitig unterstützen, die schriftliche Ausarbeitung und die Bewertung der Arbeit erfolgt aber unabhängig voneinander.

1. Arbeit: Berechnung der Jacobi-Matrix

Bei numerischen Integrationsverfahren sowie bei Stabilitätsuntersuchungen kann es erforderlich sein, die Jacobi-Matrix der Systemfunktionen während der Berechnung häufig zu aktualisieren. Hierfür müssen die Ableitungen der Systemfunktionen nach dem Zustandsvektor gebildet werden. Da diese Operationen voneinander unabhängig sind, eignen sie sich hervorragend zur Parallelisierung. Die Aufgabe umfasst die Implementierung einer parallelen Berechnung der Jacobi-Matrix, die Einbindung in die numerische Integration bzw. die Stabilitätsberechnung und die Dokumentation des Ergebnisses.

2. Arbeit: Monte-Carlo-Simulationen

Bei Monte-Carlo-Simulationen (MCS) werden Stichproben mit Hilfe vorgegebener Verteilungsfunktionen generiert und dann Berechnungen mit parametrisierten Modellen durchgeführt, um den Einfluss von streuenden Eingangsparametern auf das Systemverhalten quantifizieren zu können. Neben der direkten MCS, bei der die Stichproben unabhängig voneinander erzeugt werden können, existieren auch varianzmindernde Verfahren, die wesentlich effizienter sind, weil sie auf bereits berechneten Auswertungen zurückgreifen. Während die direkte MCS problemlos parallelisiert werden kann, sind bei den varianzmindernden Verfahren zusätzliche Kommunikationsschritte erforderlich. Die Aufgabe umfasst daher die Implementierung einer direkten und einer varianzmindernden MCSA und die Dokumentation der Effizienz.

Beginn: ab Februar 2011
Software: Matlab
Voraussetzungen: solide Grundlagen in Mathematik und Mechanik
selbstständiges Arbeiten
Spaß an Programmieraufgaben

Weitere Informationen: Prof. Dr.-Ing. Carsten Proppe (proppe@kit.edu)
Sprechstunde: montags 14:00 Uhr bis 15:00 Uhr, R. 204, Geb. 10.23