CUDA Workshop

Ausblick

Daniel Tenbrinck

Computer Vision and Pattern Recognition Group
Institut für Informatik
Westfälische Wilhelms-Universität Münster

03.Juli 2009

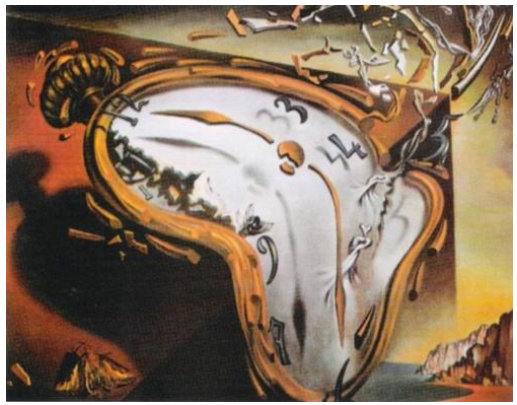
Was haben wir bisher gesehen:

Inhalt der vorangegangenen Vorträge:

- Was bedeutet Parallelisierung und wie funktioniert diese?
- Wie sind aktuelle Grafikkarten aufgebaut und wie parallelisiert man auf ihnen?
- Was ist sprachtypisch für CUDA und wie setzt man Programme um?
- Wie sehen nachvollziehbare Beispiele für CUDA-Implementierungen aus und wie schnell sind diese?
- Gibt es Anwendungen von CUDA im Umfeld des Fachbereichs und wo werden sie eingesetzt?

Ein Zitat von Walter Hesselbach:

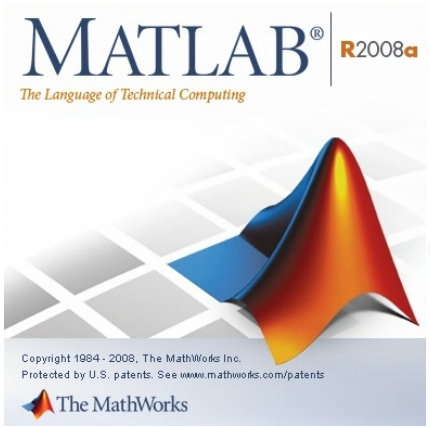
„Zukunft: die Zeit, von der man spricht, wenn man in der Gegenwart mit einem Problem nicht fertig wird.“



Was erwartet uns in der nächsten Zeit?

Fragen zur Zukunft von CUDA:

- Bleibt das Prinzip auf Nvidia Grafikkarten beschränkt?
- Werden andere Programmiersprachen Schnittstellen zu CUDA anbieten?
- Wird es möglich sein Programme unabhängig von der ausführenden Recheneinheit zu schreiben?
- Kann man in Zukunft Rechenleistung mobil machen?



CUDA-Plugin für Matlab

Wie MathWorks Matlab CUDA unterstützt:

- bisher: mex-Files liessen aus Matlab heraus C-Code ausführen

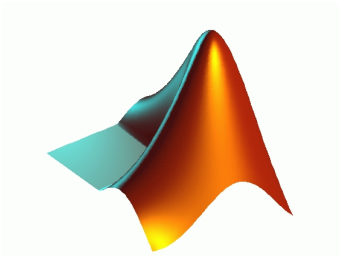
  → **Idee:** selbstgeschriebene CUDA-Plugins lassen sich aus Matlab aufrufen

- MathWorks bietet Beispiel-Quelltexte im Internet an:

  `http://developer.nvidia.com/object/matlab_cuda.html`

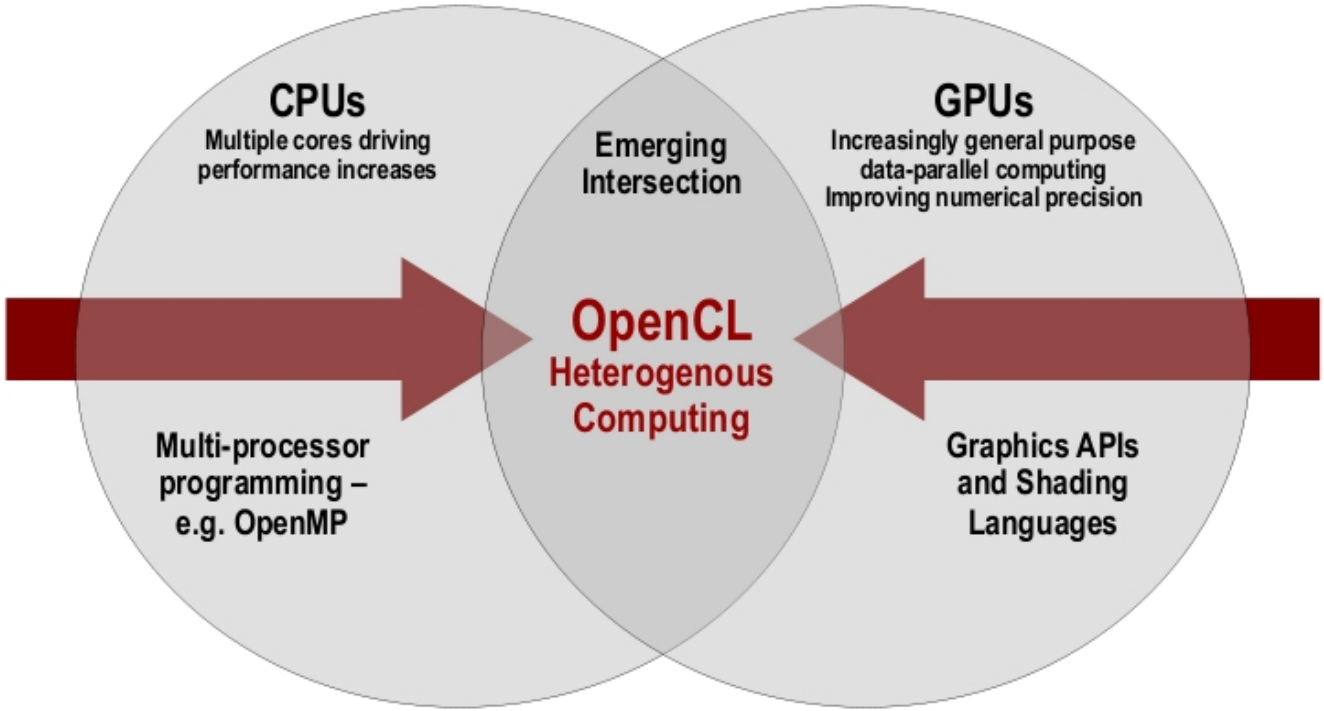
Beispiel: Schnelle Fourier-Transformation

- jeder Aufruf der FFT innerhalb von Matlab wird abgefangen
- Parameter werden an CUDA-Plugin für FFT übergeben
- Berechnung findet auf Grafikkarte statt
- Ergebnis wird der aufrufenden Funktion in Matlab übergeben



Was ist OpenCL?

- OpenCL = Open Computing Language
- Spezifikationen im Dezember 2008 von Khronos Group vorgestellt
- Programmierplattform für GPUs, CPUs und DSPs



OpenCL


KHRONOS GROUP
Over 100 companies creating authoring and acceleration standards
Board of Promoters

OpenCL

OpenCL im Detail:

- offener Standard der durch Wirtschaftsgrößen definiert wurde
- eigene zugehörige Programmiersprache: **OpenCL C** (ISO C99)
- erstmaliger Einsatz: vorr. Apple mit OS X 10.6 (Snow Leopard)
- Verteilung von Programmteilen auf mehrere Devices

  $\rightarrow$ GPU und CPU gemeinsam programmierbar!
- Programme unabhängig von Recheneinheiten

  $\rightarrow$eigene Programme werden wesentlich portabler!
- CUDA wird OpenCL zukünftig auch unterstützen
- direkter Zugriff auf OpenGL Objekte wie z.B. Texturen

CUDA auf dem Handy

Mobile Rechenleistung:

- geplant: neue Handygenerationen mit Nvidia Grafikkarten ausstatten

  → OpenCL für mobile Endgeräte ermöglicht neue Technologien
- sehr interessant im Bereich Computer Vision und Mustererkennung

Mögliche Anwendungen:

- Erkennung von wichtigen Gebäuden

  → Abruf von Informationen aus dem Internet
- Gesichtserkennung mit Handy-Kamera
- Optical Flow Berechnung in Echtzeit
- OCR mit dem Handy

  → Übersetzung von ausländischen Schriften

  → Vorlesen von Texten für Sehbehinderte

Viel Spaß mit CUDA!

Schluss