Compute

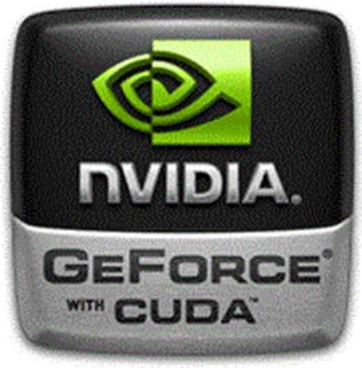
Unified

Device

Architecture

Gliederung

- Was ist CUDA?
  - CPU
  - GPU/GPGPU
  - CUDA
- Anwendungsbereiche
- Wirtschaftlichkeit
- Beispielvideo



©Nvidia

Was ist CUDA?

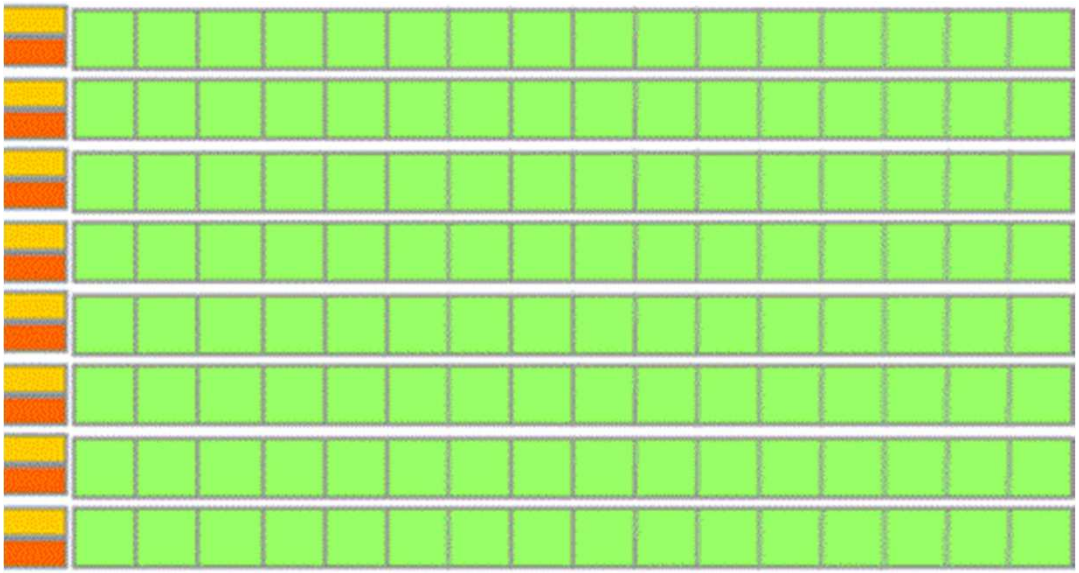
CUDA ist eine von Nvidia GPGPU-Technologie, die es Programmierern erlaubt, Programmteile zu entwickeln, die durch den Grafikprozessor (GPU) auf der Grafikkarte abgearbeitet werden. Auf der Seite der Hardware steht die GPGPU auf der Softwareseite die CUDA SDK bestehend aus eine C/C++ Compilers, Debugging Tools sowie den benötigten Libraries. Runterzuladen bei http://developer.nvidia.com/cuda-downloads
.
Um zu klären wie eine GPU bzw. die weiterentwickelte GPGPU funktioniert sehen wir uns am besten zur Veranschaulichung die CPU als Gegensatz an.

Central **P**rocessing **U**nit

Eine CPU ist für universelle Befehle ausgelegt: Sie muss mit willkürlichen Speicherzugriffen, mit Verzweigungen und einer ganzen Reihe von unterschiedlichen Datentypen zurechtkommen. Wie man im Bild sehen kann wird ein großer Teil des Chips vom Cache und dem Controller eingenommen. Die Recheneinheiten (ALU) müssen im Gegensatz dazu mit sehr wenig Platz auskommen.

Das oben verwendete Bilde wurde aus einem Bericht auf Tom´s Hardware entnommen, http://www.tomshardware.de/CUDA-Nvidia-CPU-GPU,testberichte-240065.html

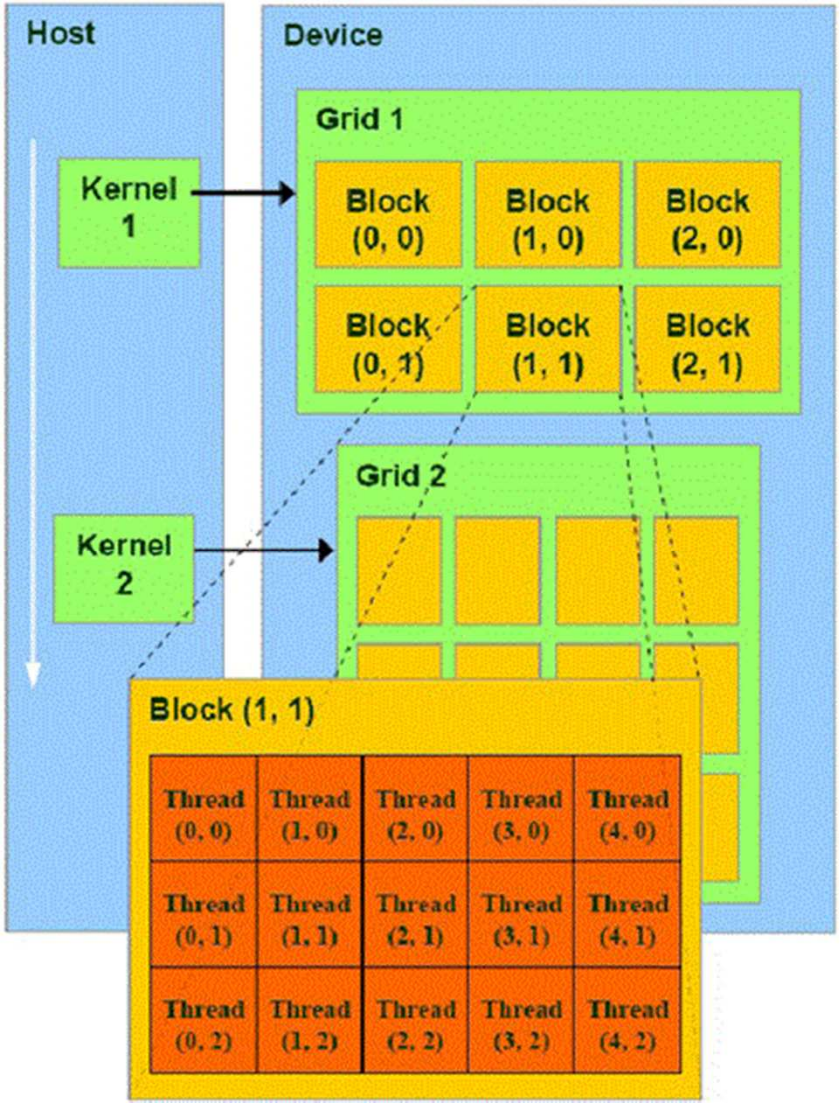
Graphics **P**rocessing **U**nit



GPUs wurden von Anfang an auf Grafikberechnungen ausgelegt .Ihre Aufgabe besteht darin Pixel zu berechnen. Der Grafikchip arbeitet nach dem SIMD-Prinzip (Single Instruction, Multiple Data und bedeutet das viele Prozessoren ihre Befehle von einem Befehlsprozessor erhalten. Ein Befehl wird also gleichzeitig auf mehrere Datensätze angewendet und parallel abgearbeitet z.B. Video, Foto Bearbeitung) wodurch die ALUs besser ausgenutzt werden können. Da die Pixel im Speicher nebeneinander liegen, wird nur ein kleiner Cache benötigt und die Zugriffszeiten fallen wesentlich geringer aus.

Das oben verwendete Bilde wurde aus einem Bericht auf Tom´s Hardware entnommen, http://www.tomshardware.de/CUDA-Nvidia-CPU-GPU,testberichte-240065.html

CUDA (hardwareseitig)



- CPU lädt in den Kernel auf der GPU
- Threads gehören zu einem Block
- Blöcke gehören zu einem Grid

- Für die genau Zuweisung durch den Programmierer hat jeder Thread, Block, Grid und Kernel eine eigene ID

Das oben verwendete Bilde wurde aus http://www.nvidia.com/content/CUDA-ptx_isa_1.4.pdf entnommen , © Nvidia

CUDA (softwareseitig)

SDK besteht aus einer Erweiterung für C sowie einer speziellen Laufzeit-Bibliothek

„C“ Erweiterung wird in 4 Bereiche unterteilt:
•Funktions-Modifikatoren – Damit wird angegeben ob Funktionen auf der GPU oder der CPU ausgeführt werden

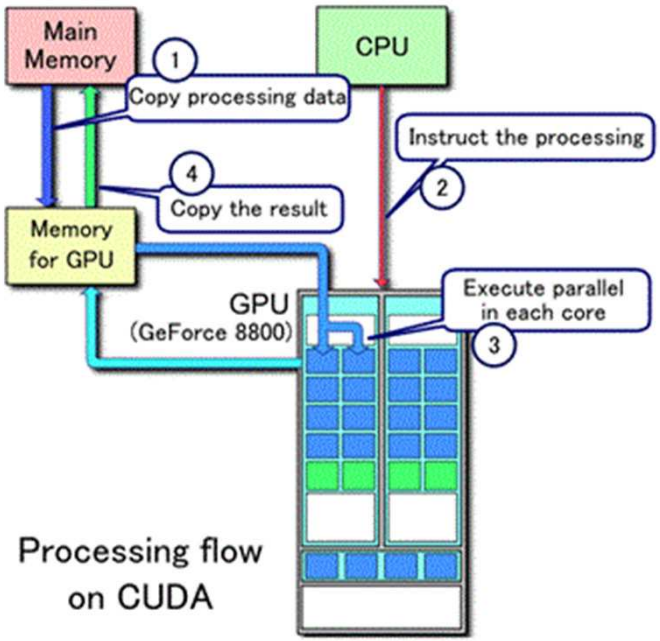
•Variablen-Modifikatoren – Damit wird angegeben in welchem Speicher auf der GPU die Variablen abgelegt werden sollen

•Befehl zum Initalisieren der einzelnen Elemente

•4 native Variablentypen – um Block- und Gridgrößen, sowie Block- und Gridindizes anzugeben

Neben C kann man sich allerdings auch noch in andere Programmiersprachen wagen um CUDA zu programmieren, z.B. Fortran was besonders bei Wissenschaftlern gefragt ist, „OpenCL“ (Open Computing Language) eine ursprünglich von Apple entwickelte Programmiersprache und „DirectCompute“ welches von Microsoft entwickelt wurde und fester Bestandteil von Directx11 ist.

Kompiliervorgang von CUDA



Die Software welche mit in CUDA implementiert wird muss aus zwei Teilen bestehen. Einmal dem Host welcher die Ausführung des Programms steuert, die Daten bereitstellt und die Ergebnisse zur Weiterverarbeitung entgegen nimmt. Und dem Kernel der den eigentlichen Algorithmus enthält und diesen nun parallel in der GPU abarbeitet.

Das oben verwendete Bilde wurde von
http://en.wikipedia.org/wiki/File:CUDA_processing_flow_(En).PNG entnommen, by Tosaka

Anwendungsbereiche

•**Foto –und Videobearbeitung** – für den Heimgebrauch, Badaboom um Filme von DVD oder Bluray auf das IPhone oder andere mobilen Geräte zu konvertieren. Aber auch um Videos in Echtzeit von einem verwackelten unscharfen Bild zu einem schaubarem Video zu konvertieren.

•**Medizin** – um CT - Bilder klarer darzustellen und das Erkennen von Anomalien zu vereinfachen. Bisher war dieses Verfahren für den klinischen Alltag durch die hohe Ausführungszeit (ca. 9 min für ein einzelnes Bild bei 80 Durchgängen) nicht geeignet - Durch GPU Computing wird die Technik nutzbar gemacht. Fast 2300 mal schneller als mit einer CPU.

•**Finanzen -** Mit nur 12 CUDA – fähigen Grafikprozessoren ist Hanwecks Software Volera in der Lage den gesamten US-amerikanischen Optionsmarkt in Echtzeit zu berechnen, dafür waren bisher über 60 konventionelle Server benötigt wurden.

•**Neue Energien -** z.b. zur Auswertung von seismischen Daten um Ölbohrungen zu Unterstützen.

Wirtschaftlichkeit

Eine Tesla Workstation wie man sie oben sieht kostet rund 10.000 Dollar und beinhaltet 4 Tesla C1060 GPUs, sie benötigt für die rekonstruktion einer digitalen Thomographie: 59 Sekunden. Um das selbe mit einer bzw. mehreren CPU´s zu erreichen ist ein vielfaches des Geldes nötig.

Quellen:

- Winfwiki
- Wikipedia
- Elektronik Kompendium
- Nvidia
- Tom´s Hardware