LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

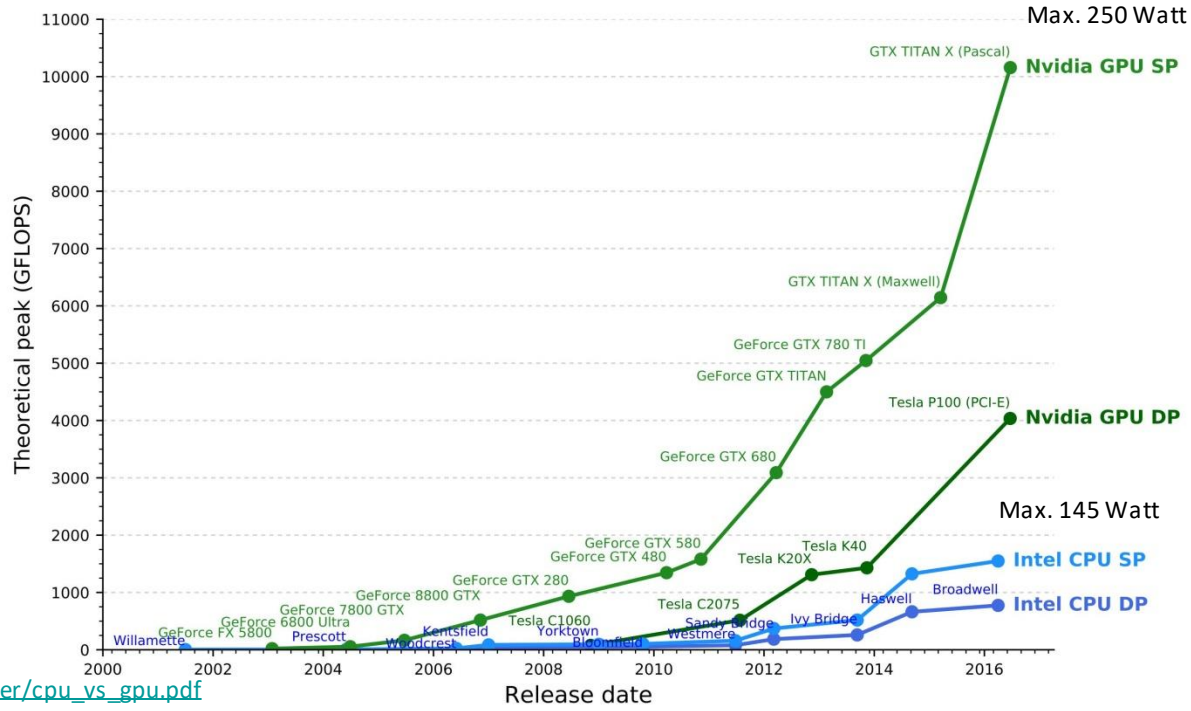
Dominik Bitzer

General-purpose computing on GPU (GPGPU)

- Blockpraktikum Modern Computer Architecture
- Aufgabensteller: Dr. Gordon Cichon
- Datum: 01. August 2018

Motivation und Hintergrund

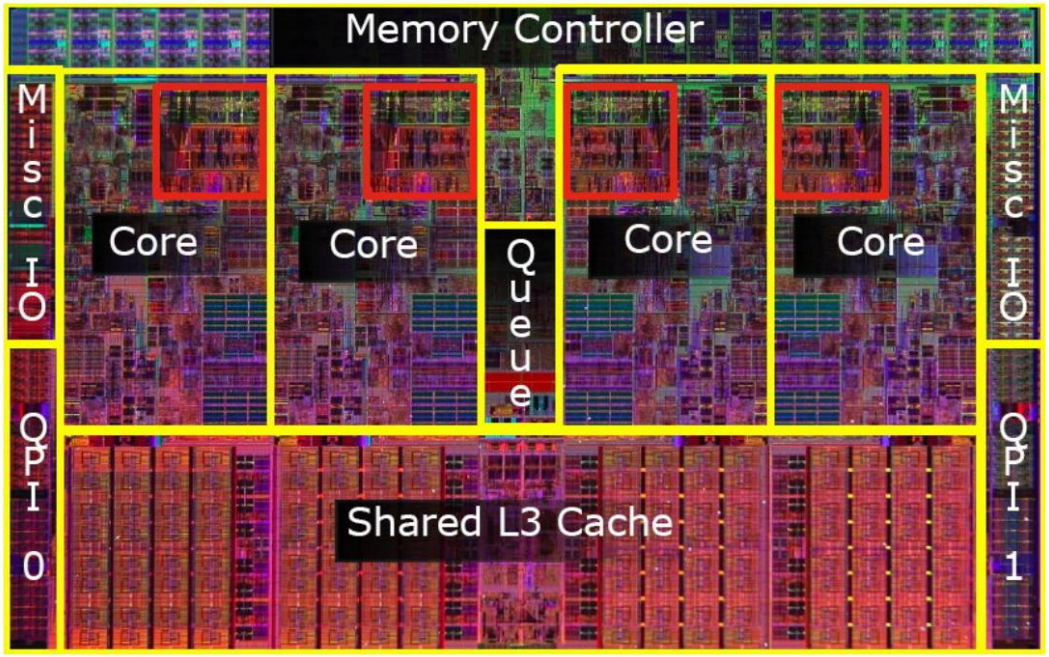
- GPU vs. CPU:
  - GLOPs: ca. 6,5x
  - Bandbreite: ca. 4,3x
  - Leistungsaufnahme: ca. 1,7x
  - Effizienz: ca. 3,8x (GFLOP / Watt)



Quellen:
https://github.com/mgalloy/cpu-vs-gpu/blob/master/cpu_vs_gpu.pdf
https://ark.intel.com/products/91317/Intel-Xeon-Processor-E5-2699-v4-55M-Cache-2_20-GHz
https://www.geforce.com/hardware/desktop-gpus/geforce-gtx-titan-x/specifications

Motivation und Hintergrund

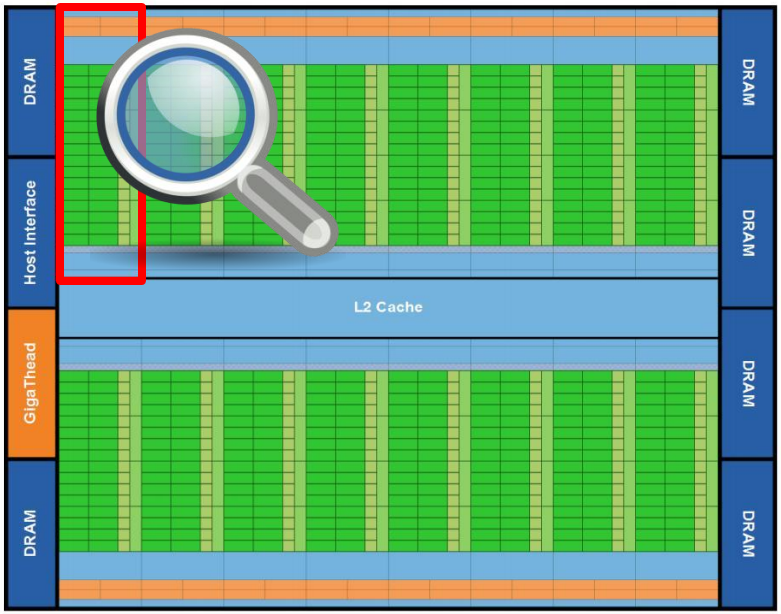
- CPU Architektur:
  - Wenige Kerne mit hoher single-thread Leistung
  - Wenige Register, aber große, hierarchisch organisierte Caches
  - Ziel: geringe Latenz, schnelle Abarbeitung serieller Tasks

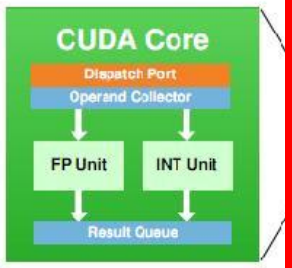


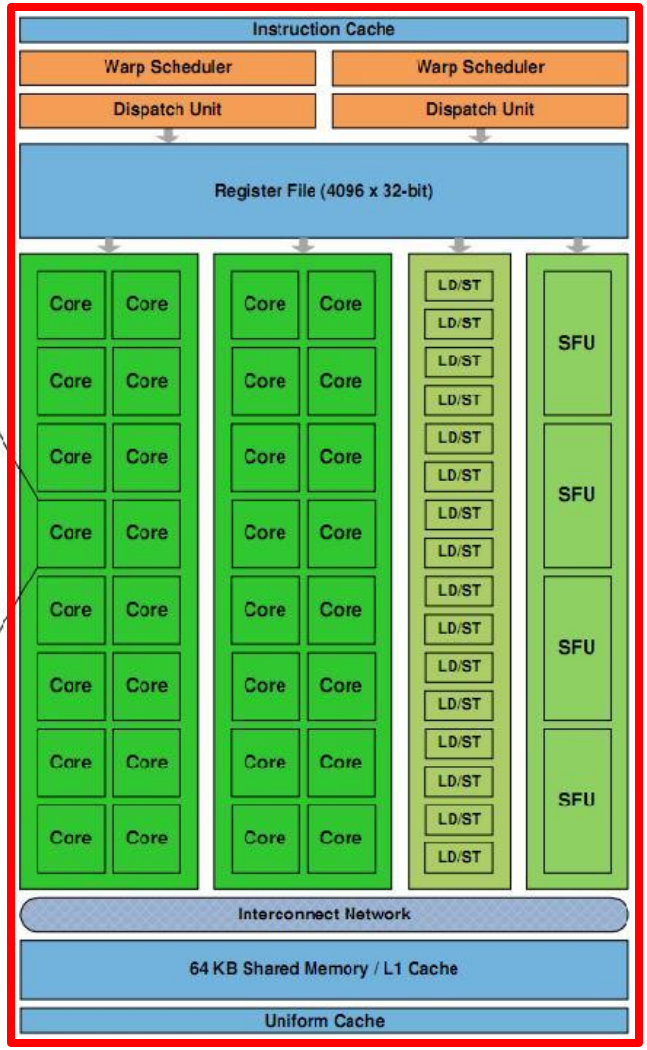
Quellen:
P. Jonathan and T. Josh: A Comparison of Modern GPU and CPU Architectures: And the Common Convergence of Both, 2011

Motivation und Hintergrund

- GPU Architektur:
  - Viele Kerne mit geringer single-thread Leistung
  - Viele Register, kleinere Caches
  - Ziel: hoher Durchsatz ähnlicher Berechnungen







Quellen:

P. Jonathan and T. Josh: A Comparison of Modern GPU and CPU Architectures: And the Common Convergence of Both, 2011

Projekt: Netflix Korrelationen

- Netflix Price: öffentlicher Datamining Wettbewerb
- >100 Mio. Bewertungen von 17.000 Filmen durch ca. 500.000 Nutzer
- Idee: Korrelationsmatrix über Filme anhand Bewertungen

Bewertung Film x durch Benutzer i

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

(Definition Pearson's Correlation)

$$r_{xy} = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\ \sqrt{n\sum y_i^2 - (\sum y_i)^2}}$$

(gleich, nur programmiererfreundlich)

Quellen:
https://www.mathsisfun.com/data/correlation.html
https://www.kaggle.com/netflix-inc/netflix-prize-data

GPGPU Herausforderungen

Portierung von CPU Code zu GPGPU Code:

- GPU kann keine CPU-Funktionen aufrufen, darunter fallen z.B. nicht-portierte STL-Funktionen (siehe Vortrag Thomas)

Weitere eventuelle Probleme:

- Technische Einschränkungen, z.B. Double Precision Leistung wesentlich niedriger als Single Precision (aber z.B. Lösung durch Nvidia Tesla)
- Evtl. Lock-In auf Anbieter, CUDA-Code nicht auf AMD-GPUs nutzbar (OpenCL bietet hierfür Lösung)

GPGPU Herausforderungen

GPU-Parallelismus muss genutzt werden, z.B. Schleifen umschreiben

Single-Thread Schleife

```
for (int i = 0; i < n; i++)
    z[i] = x[i] + y[i];
```

x	6	8	2	8	7	6	3	2	5	5	3	2	9	10	9	0	7	4	4	3
	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
y	8	1	6	3	7	9	8	7	4	8	4	10	4	8	1	1	10	10	1	3
	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=
z	14	9	8	11	15	15	11	10	9	13	7	12	13	18	10	1	17	13	5	6

Quellen:
https://devblogs.nvidia.com/even-easier-introduction-cuda/

GPGPU Herausforderungen

GPU-Parallelismus muss genutzt werden, z.B. Schleifen umschreiben

Single-Thread Schleife

```
for (int i = 0; i < n; i++)
    z[i] = x[i] + y[i];
```

Multi-Thread Schleife

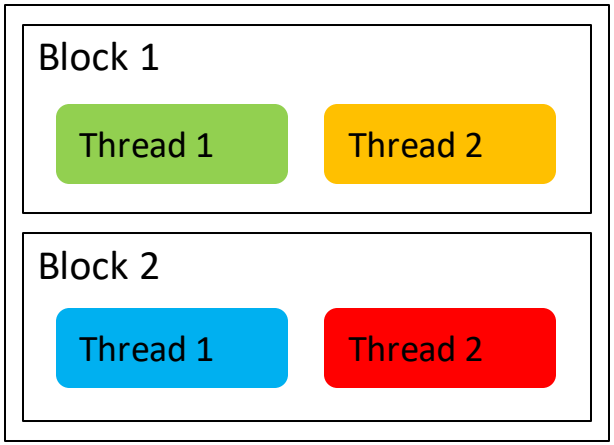
```
int index = blockIdx.x * blockDim.x + threadIdx.x;
int stride = blockDim.x * gridDim.x;
for (int i = index; i < n; i += stride)
    z[i] = x[i] + y[i];
```

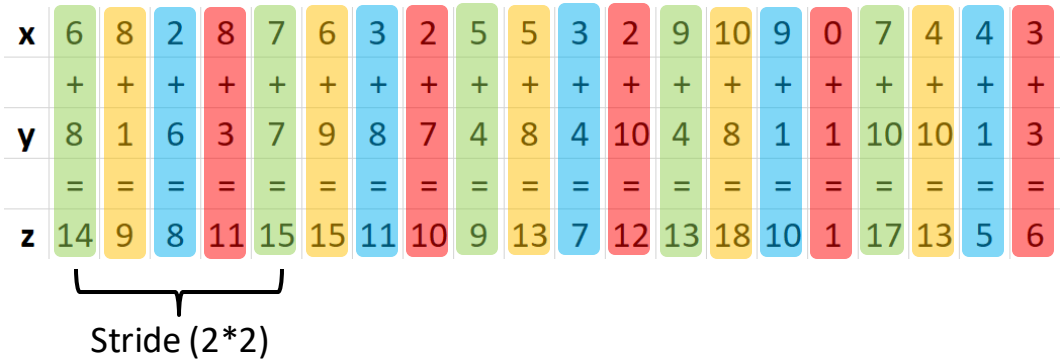
Block-Position in Grid

Block-Größe

Thread-Position in Block

Grid-Größe





Quellen:
https://devblogs.nvidia.com/even-easier-introduction-cuda/

GPGPU Herausforderungen

Beschränkung Kommunikation Host und GPU durch Latenz und Bandbreitebeschränkungen des Systembus

z.B. GPU-Speicher zu GPU 144 GB/s (NVIDIA Tesla C2050)
vs. Hauptspeicher zu GPU 8 GB/s (PCIe x16 Gen2)

Datenaustausch zwischen muss optimiert werden, z.B.:

- Batching von Datentransfers
- Parallelisierung von Berechnungen und Transfers
- GPU für möglichst großen Teil des Programms verwenden

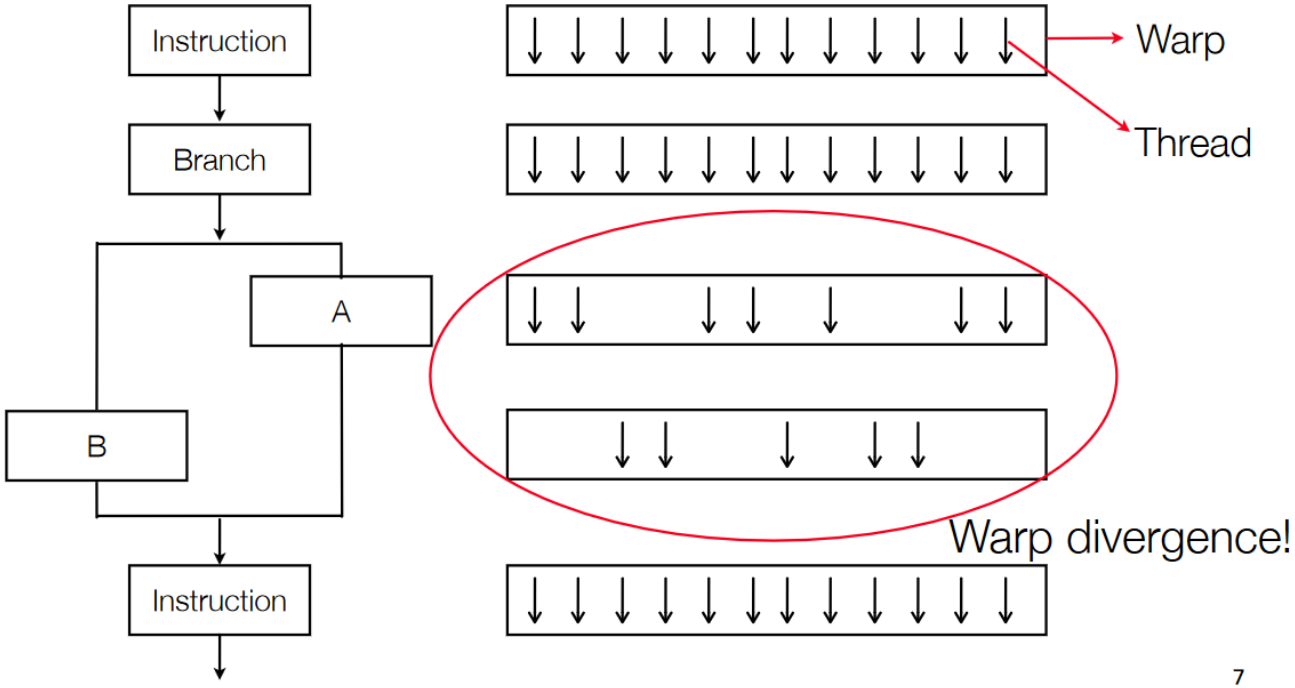
```
A_global_mem = cuda.to_device(np.array([1, 2, 3]))
B_global_mem = cuda.to_device(np.array([4, 5, 6]))
C_global_mem = cuda.device_array((3, 2))  # result array
[…]
# Start the kernel
cuda_kernel[blockspergrid, threadsperblock](A_global_mem, B_global_mem, C_global_mem)
res = C_global_mem.copy_to_host()
```

Quellen:
https://devblogs.nvidia.com/how-optimize-data-transfers-cuda-cc/

GPGPU Herausforderungen

Alle CUDA Threads in Wrap (Teil von Block) arbeiten Instruktionen gleichzeitig ab

Jeder Thread geht jeden Zweig durch (Branch Divergence)

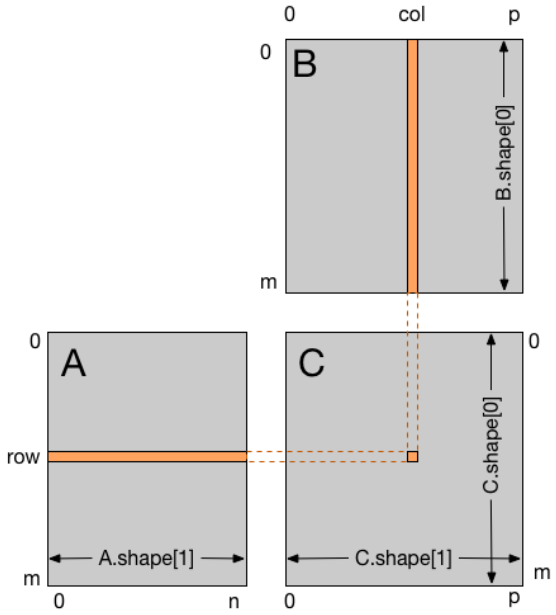


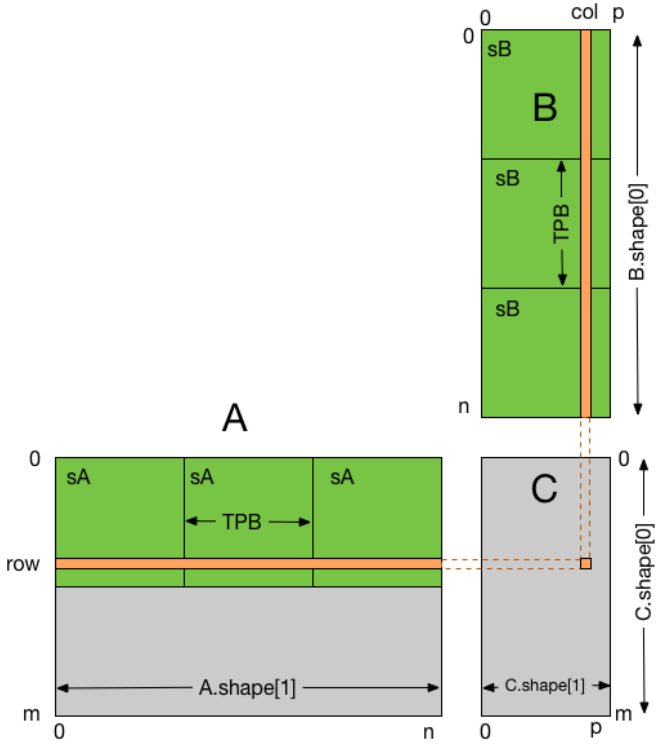
Quellen:
https://www.bu.edu/pasi/files/2011/07/Lecture4.pdf

GPGPU Herausforderungen

Caches von GPU-Prozessoren sehr klein, häufige Cache-Misses

Optimierung, explizites Laden von Daten aus (GPU-) Global Memory in schnelleren „Shared Memory“





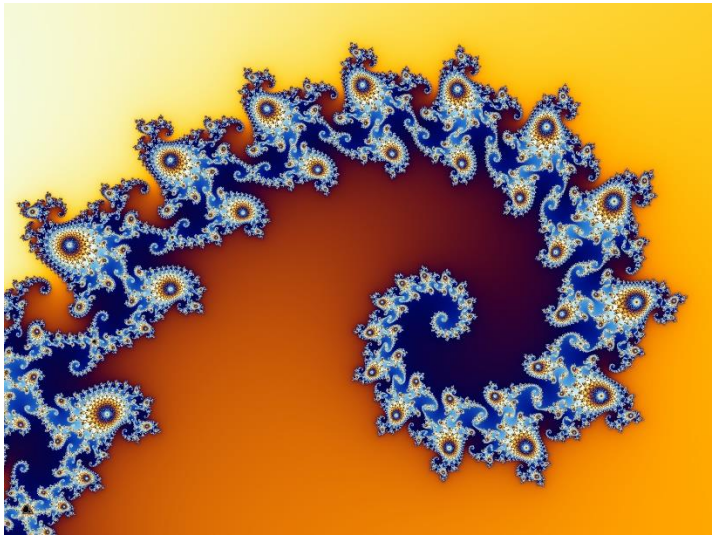
Quellen:
https://nyu-cds.github.io/python-numba/05-cuda/

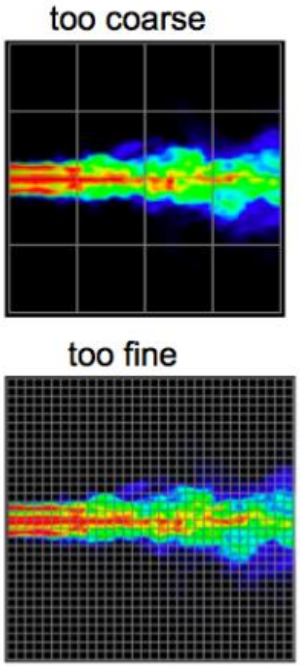
GPGPU Herausforderungen

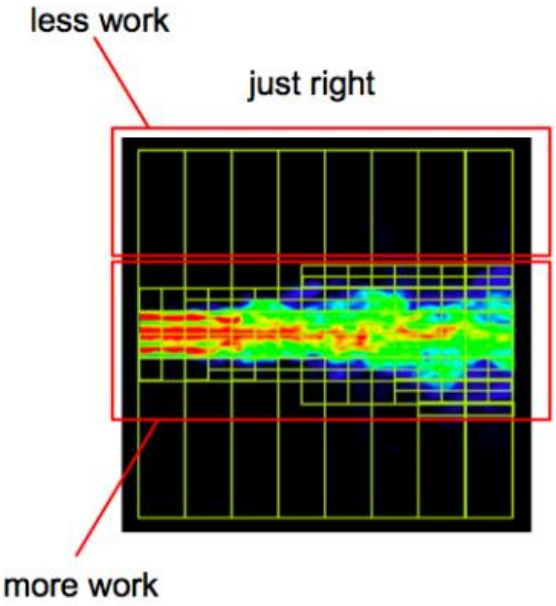
Aufteilung eines Problems auf große Anzahl Threads

Gleichmäßige Aufteilung von Rechenlast z.B. bei Simulationen schwer

Verbesserung der Auslastung z.B. durch „Dynamic Parallelism“







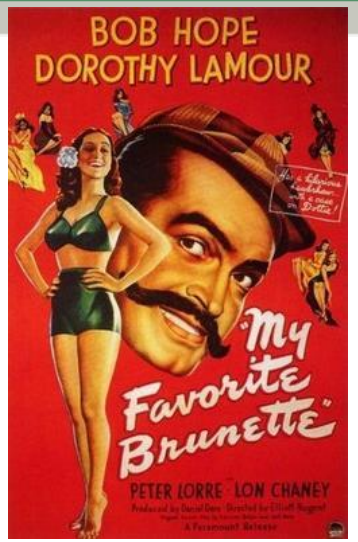
Quellen:
https://devblogs.nvidia.com/introduction-cuda-dynamic-parallelism/
http://www.misterx.ca/Mandelbrot_Set/M_Set-IMAGES_&_WALLPAPER.html

Ergebnisse der Berechnungen

Maximale gefundene Korrelation (0,37):
My Favorite Brunette (1947 American romantic comedy) und
The Lemon Drop Kid (1951 comedy film)

Minimale gefundene Korrelation (-0,124):

Was das Herz begehrt (mit Jack Nicholson) und
Aqua Teen Hunger Force (Cartoon-Serie)



	Dinosaur	Isle of Ma	Character	Paula Abd	The Rise	Sick	8 Man	What the			My Favor	Lord of th	Nature: A
Dinosaur Planet		0,034	0,01	0,031	0,029	0,015	0,039	0,005			0,034	0,01	0,038
Isle of Man TT 2004 Review			0,016	0,041	0,047	0,019	0,068	0,005			0,023	0,007	0,053
Character				0,018	0,009	0,032	0,023	0,02			0,022	0,003	0,01
Paula Abdul's Get Up & Dance					0,015	0,022	0,052	0,005			0,017	0,007	0,038
The Rise and Fall of ECW						0,012	0,038	-0,007	0,035	0,016	0,021	0,011	0,028
Sick							0,028	0,028	0,037	0,065	0,038	0,018	0,004	0,016
8 Man								0,004	0,074	0,058	0,036	0,043	0,009	0,038
What the #$*! Do We Know!?									0,007	0,01	0,014	0,006	0,001	0,007
Class of Nuke 'Em High 2										0,032	0,021	0,025	0,018	0,037
Fighter											0,057	0,025	0,01	0,028
Full Frame: Documentary Shorts												0,017	-0,001	0,032
My Favorite Brunette													0,002	0,03
Lord of the Rings: The Return of the King: Extended Edition: Bonus Material														-0,001