PDC Summer School 2016

CUDA developer tools

2015-08-19

Michael Schliephake

Szilárd Páll

KTH – CSC – HPCViz

Compiler

- Compiler toolchain
- Host – device compilation

- Device code generation
  - Multi-pass
  - Allows fallback code-path
  - Embed multiple code versions (cirtual vs physical architecture)

Memory checker

- Cuda-memcheck
- Checks for:
  - Misaligned/out of bounds access
  - Device malloc()/free()
  - API errors, hardware errors
  - Memory leaks
  - races

Debugger

- Cuda-gdb
- Nsight: VS or Eclipse plugin
- Parallel debuggers:
  - Totalview
  - Allinea DDT

Profilers

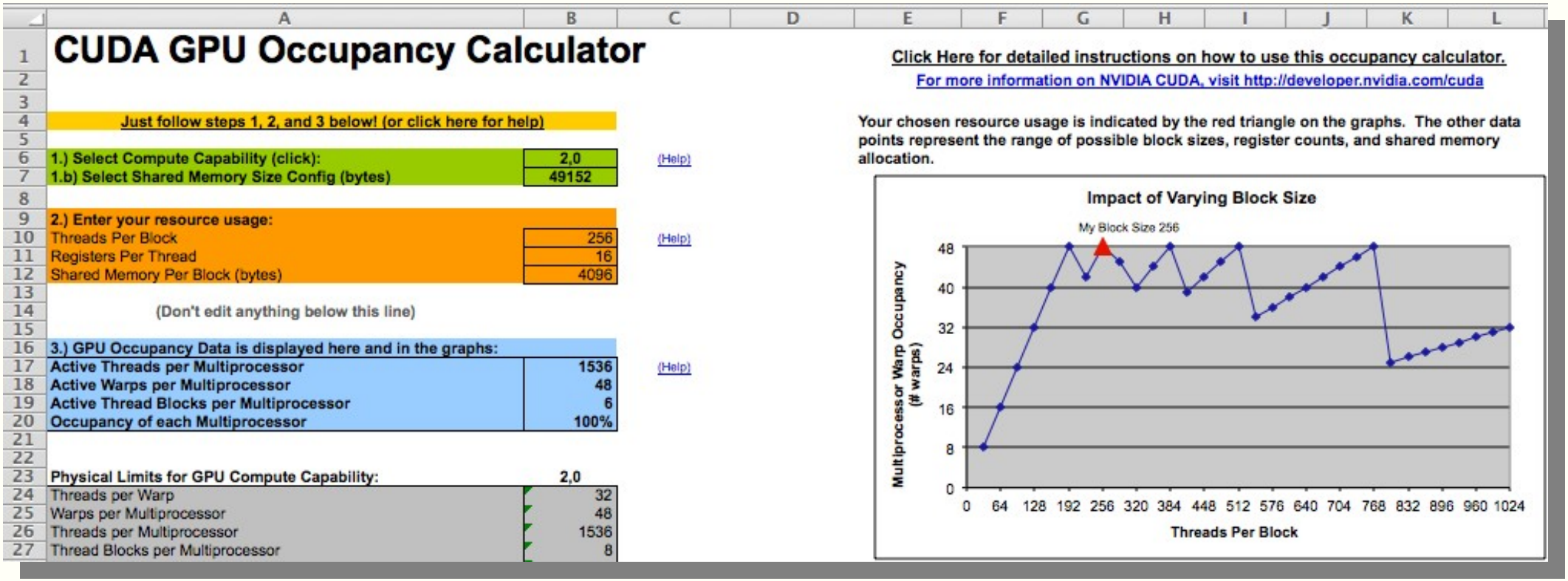
- nvprof
  - Command line tool for device profiling
  - API tracing
  - Can generate data for nvvp
- Visual Profiler (nvvp)
  - Linux/OS X
- Nsight (VS)
  - Visual Studio plugin
- 3rd party tools: TAU, Vampirtrace, PAPI

GPU management & monitoring

- nvidia-smi
  - Monitor GPU at runtime
  - Manage behavior (e.g. clocks, ECC)
- NVML
  - Programatically do the same (and more)

CUDA Occupancy Calculator

http://developer.download.nvidia.com/compute/cuda/CUDA_Occupancy_calculator.xls



CUDA GPU Occupancy Calculator

Click Here for detailed instructions on how to use this occupancy calculator.
For more information on NVIDIA CUDA, visit http://developer.nvidia.com/cuda

Just follow steps 1, 2, and 3 below! (or click here for help)

1.) Select Compute Capability (click):	2.0	(Help)
1.b) Select Shared Memory Size Config (bytes)	49152	
2.) Enter your resource usage:		
Threads Per Block	256	(Help)
Registers Per Thread	16	
Shared Memory Per Block (bytes)	4096	

(Don't edit anything below this line)

3.) GPU Occupancy Data is displayed here and in the graphs:		
Active Threads per Multiprocessor	1536	(Help)
Active Warps per Multiprocessor	48	
Active Thread Blocks per Multiprocessor	6	
Occupancy of each Multiprocessor	100%	

Physical Limits for GPU Compute Capability:	2.0
Threads per Warp	32
Warps per Multiprocessor	48
Threads per Multiprocessor	1536
Thread Blocks per Multiprocessor	8

Your chosen resource usage is indicated by the red triangle on the graphs. The other data points represent the range of possible block sizes, register counts, and shared memory allocation.