CUDA KERNEL PROFILING USING NVIDIA NSIGHT COMPUTE

Sanjiv Satoor, Magnus Strengert
AGENDA

Overview of NVIDIA’s Devtools

What is New for CUDA Tools in CUDA 10.1

Working with NVIDIA Nsight Compute (Demo)
TOOLS OFFERINGS

IDE
- Nsight Eclipse Edition (EE) (plugin)
- Nsight Visual Studio Edition (VSE) (plugin)

Debug
- Nsight
- CUDA-GDB
- CUDA Debug API

Memcheck
- CUDA-memcheck
- Nsight Visual Studio Edition built-in

Profile
- Nsight Systems
- Nsight Compute
- CUDA Visual Profiler
- nvprof
- Nsight Visual Studio Edition
- CUPTI

NVTX (NVidia Tools eXtension)
CUDA PROFILING TOOLS
Updates for CUDA 10.1

General:
- Support for latest Turing GPUs

Visual Profiler/nvprof/CUPTI:
- Support for NVTX string registration API nvtxDomainRegisterStringA().

Nsight Visual Studio Edition:
- Nsight Compute improvements
Updates:

- Lower performance overhead and reduced memory overhead
- Source Page reports metrics by functions or files
- Updates sections and rules; section descriptions

Improved Parity to NVPROF:
- Profiling of child processes (MPI, ...)
- CLI options: --summary, --quite, ...

Extended NVTX Support:
- Trigger profiling by NVTX ranges
- Print NVTX state in CLI
NSIGHT PRODUCT FAMILY

Standalone Performance Tools

Nsight Systems - System-wide application algorithm tuning
Nsight Compute - Debug/optimize specific CUDA kernel
Nsight Graphics - Debug/optimize specific graphics shader

IDE Plugins

Nsight Eclipse Edition/Visual Studio - editor, debugger, some perf analysis
System-wide application algorithm tuning
  Multi-process tree support

Locate optimization opportunities
  Visualize millions of events on a very fast GUI timeline
  Or gaps of unused CPU and GPU time

Balance your workload across multiple CPUs and GPUs
  CPU algorithms, utilization, and thread state
  GPU streams, kernels, memory transfers, etc
TRANSITIONING TO PROFILE A KERNEL

Dive into kernel analysis
NVIDIA NSIGHT COMPUTE
Next-Gen Kernel Profiling Tool

Key Features:
• Interactive CUDA API debugging and kernel profiling
• Graphical profile report
• Comparison of multiple kernel reports
• Fully Customizable (Reports and Analysis Rules)
• Command Line, Standalone, IDE Integration

OS: Linux, Windows, ARM, MacOSX (host only)

GPUs: Pascal (GP10x), Volta, Turing

NSIGHT COMPUTE DEMO
**WARP SCHEDULER**

**Volta Architecture**

- 4 Warp Scheduler per SM
- Manages a pool of warps:
  - Volta: 16 warp slots
  - Turing: 8 warp slots
- Each scheduler can issue 1 warp/cycle
- Offers simplified mental model for profiling and SM metrics
WARP SCHEDULER
Mental Model for Profiling

Warp States:
- Unused
- Active
- Stalled
- Eligible
- Selected

Warp Slots
7 6 5 4 3 2 1 0
WARP SCHEDULER
Mental Model for Profiling

Cycle: N
Each Cycle:
Out of all eligible warps, select one to issue on that cycle
WARP SCHEDULER
Mental Model for Profiling

Warp States:
- Unused
- Active
- Stalled
- Eligible
- Selected

Cycle: N

Each Cycle:
Out of all eligible warps, select one to issue on that cycle

Warp Slots

Issue Slot: 3
WARP SCHEDULER
Mental Model for Profiling

Warp States:
- Unused
- Active
- Stalled
- Eligible
- Selected

Cycle: N, N+1

Warp selected in cycle N, is not eligible in N+1.
E.g. instructions with longer instruction latencies

Issue Slot: 3, 4
WARP SCHEDULER
Mental Model for Profiling

Warp States:
- Unused
- Active
- Stalled
- Eligible
- Selected

Cycle:
- N
- N+1
- N+2

Warp Slots

Issue Slot: 3 4 X
No eligible warp! Issue slot unused
WARP SCHEDULER

Mental Model for Profiling

Warp States:
- Unused
- Active
- Stalled
- Eligible
- Selected

Cycle: N  N+1  N+2  N+3

Warp Slots

Issue Slot: 3 4 2

New warps scheduled
WARP SCHEDULER

Mental Model for Profiling

Warp States:
- Unused
- Active
- Stalled
- Eligible
- Selected

Cycle:
- N
- N+1
- N+2
- N+3

Metrics (aggregated):
WARP SCHEDULER
Mental Model for Profiling

Warp States:
- Unused
- Active
  - Stalled
  - Eligible
  - Selected

Warp Slots
- 7
- 6
- 5
- 4
- 3
- 2
- 1
- 0

Cycle:
N+1
N+2
N+3

Metrics (aggregated):
cycles_active

4
WARP SCHEDULER
Mental Model for Profiling

Warp States:
- Unused
- Active
- Stalled
- Eligible
- Selected

Cycle:
- Warp Slots

Issue Slot:
- Issue Slot: 3 4 X 2

Metrics (aggregated):
- cycles_active: 4
- warps_active: 20
WARP SCHEDULER
Mental Model for Profiling

Warp States:
- Unused
- Active
- Stalled
- Eligible
- Selected

Cycle:
- N
- N+1
- N+2
- N+3

Metrics (aggregated):
- cycles_active: 4
- warps_active: 20
- warps_active/cycles_active: 5
- achieved_occupancy: 62.5%
WARP SCHEDULER
Mental Model for Profiling

Warp States:
- Unused
- Active
- Stalled
- Eligible
- Selected

Cycle:
- N
- N+1
- N+2
- N+3

Issue Slot:
- Issue Slot 3
- Issue Slot 4
- Issue Slot 2

Metrics (aggregated):  
cycles_active 4  
warps_active 20  
warps_active/cycles_active 5  
achieved_occupancy 62.5%  
warp_stalled 15
WARP SCHEDULER
Mental Model for Profiling

Warp States:
- Unused
- Active
- Stalled
- Eligible
- Selected

Cycle:
- N
- N+1
- N+2
- N+3

Metrics (aggregated):
- cycles_active: 4
- warps_active: 20
- warps_active/cycles_active: 5
- achieved_occupancy: 62.5%
- warps_stalled: 15
- warps_eligible: 5

Issue Slot:
- Issue Slot: 3, 4, X, 2
WARP SCHEDULER
Mental Model for Profiling

Warp States:
- Unused
- Active
- Stalled
- Eligible
- Selected

Cycle: N N+1 N+2 N+3

Metrics (aggregated):
- cycles_active: 4
- warps_active: 20
- warps_active/cycles_active: 5
- achieved_occupancy: 62.5%
- warps_stalled: 15
- warps_eligible: 5
- warps_issued: 3

Issue Slot: [3, 4, X, 2]
WARP SCHEDULER
Mental Model for Profiling

Warp States:
- Unused
- Active
- Stalled
- Eligible
- Selected

Cycle:  N  N+1  N+2  N+3

Warp Slots:

Metrics (aggregated):
- cycles_active: 4
- warps_active: 20
- warps_active/cycles_active: 5
- achieved_occupancy: 62.5%
- warps_stalled: 15
- warps_eligible: 5
- warps_issued: 3
- warps_issued/cycles_active: 0.75
- issue_slot_utilization: 75%
**WARP SCHEDULER**

Application to kernel_B

Warp States:
- Unused
- Active
- Stalled
- Eligible
- Selected

Metrics (theoretical; every 8 cycles):
- warps_active: 8 / 1
- warps_stalled: 7 / 7/8
- warps_eligible: 1 / 1/8
- warps_selected: 1 / 1/8
WARP SCHEDULER
Application to kernel_B

Warp States:
- Unused
- Active
- Stalled
- Eligible
- Selected

Metrics (theoretical; every 8 cycles):
- warps_active: 8 / 1
- warps_stalled: 7 / 7/8
- warps_eligible: 1 / 1/8
- warps_selected: 1 / 1/8

Metrics (from report):
- warps_active: 1.00
- warps_stalled: 0.87
- warps_eligible: 0.13
- warps_selected: 0.13
WARP SCHEDULER
Application to kernel_A

Warp States:
- Unused
- Active
- Stalled
- Eligible
- Selected

FP64 Pipeline on Volta:
- Dependent Issue Rate: 8 cycles
- Issue Rate: 4 cycles

N  N+1  N+2  N+3  N+4  N+5  N+6  N+7  N+8
WARP SCHEDULER
Application to kernel_A

Warp States:
- Unused
- Active
- Stalled
- Eligible
- Selected

FP64 Pipeline on Volta:
- Dependent Issue Rate: 8 cycles
- Issue Rate: 4 cycles
WARP SCHEDULER
Application to kernel_A

Warp States:
- Unused
- Active
- Stalled
- Eligible
- Selected

N  N+1  N+2  N+3  N+4  N+5  N+6  N+7  N+8

FP64 Pipeline on Volta:
Dependent Issue Rate: 8 cycles
Issue Rate: 4 cycles
WARP SCHEDULER

Application to kernel_A

Warp States:
- Unused
- Active
- Stalled
- Eligible
- Selected

FP64 Pipeline on Volta:

- Dependent Issue Rate: 8 cycles
- Issue Rate: 4 cycles
WARP SCHEDULER

Application to kernel_A

Warp States:
- Unused
- Active
- Stalled
- Eligible
- Selected

FP64 Pipeline on Volta:
- Dependent Issue Rate: 8 cycles
- Issue Rate: 4 cycles
WARP SCHEDULER
Application to kernel_A

Warp States:
- Unused
- Active
- Stalled
- Eligible
- Selected

Metrics (theoretical; every 8 cycles):
- warps_active: 96
- warps_stalled: 74
- warps_eligible: 22
- warps_selected: 2

N | N+1 | N+2 | N+3 | N+4 | N+5 | N+6 | N+7 | N+8
---|-----|-----|-----|-----|-----|-----|-----|-----
Unused | Active | Stalled | Eligible | Selected | Unused | Active | Stalled | Eligible | Selected | Unused | Active | Stalled | Eligible | Selected | Unused | Active | Stalled | Eligible | Selected | Unused | Active | Stalled | Eligible | Selected

35
WARP SCHEDULER
Application to kernel_A

Warp States:
- Unused
- Active
- Stalled
- Eligible
- Selected

Metrics (theoretical; every 8 cycles):
- warps_active: 96, 96/8 = 12.00
- warps_stalled: 74, 74/8 = 9.25
- warps_eligible: 22, 22/8 = 2.75
- warps_selected: 2, 2/8 = 0.25

Metrics (from report):
- warps_active: 12.27
- warps_stalled: 9.08
- warps_eligible: 2.80
- warps_selected: 0.26
TAKEAWAYS
Profiling with Nsight Compute

General:

- Use tools whenever possible
- Understanding app/gpu/system and tools go hand in hand
  (on us to make this as easy as possible)

Nsight Compute:

- Use SOL metrics to understand overall limiter
  (reading real-world reports is a lot more difficult; but the very same workflow applies)
- Read rules’ output for guidance throughout the report
  (we’ll add more rules in future releases)
- Use top-down approach; no need to jump directly into SASS code
- Do not optimize stalls, if you already use all/sufficient issue slots
- Let us know what works for you and what doesn’t:
  devtalk.nvidia.com > Development Tools > Nsight Compute
DEVELOPER TOOLS AT GTC19

Talks:
S9751: Accelerate Your CUDA Development with Latest Debugging and Code Analysis Developer Tools, Tue @9am
S9866: Optimizing Facebook AI Workloads for NVIDIA GPUs, Tue @9am
S9345: CUDA Kernel Profiling using NVIDIA Nsight Compute, Tue @1pm
S9661: Nsight Graphics - DXR/Vulkan Profiling/Vulkan Raytracing, Wed @10am
S9503: Using Nsight Tools to Optimize the NAMD Molecular Dynamics Simulation Program, Wed @1pm

Hands-on labs:
L9102: Jetson Developer Tools Training Lab, Mon @9am, 11:30am
L9124: Debugging and optimizing CUDA applications with Nsight products on Linux training lab, Tue @8am, 10am

Connect with the Experts (where Developer Tools will be available):
CE9123: CUDA & Graphics Developer Tools, Tue @2pm, Wed @3pm
CE9137: Jetson Embedded Platform, Tue @12pm, 5pm, Wed @1pm, 4pm, Thu @12pm

Podium: Demos of DevTools products on Linux, DRIVE AGX & Jetson AGX at the showfloor
  Tue @12pm - 7pm
  Wed @12pm - 7pm
  Thu @11am - 2pm
BACKUP SLIDES
CUDA PROFILING TOOLS

NVIDIA® Visual Profiler - CUDA trace/kernel profiling
  GUI - runs on Linux, Windows, Mac

nvprof - CUDA trace/kernel profiling
  Command line tool - runs on Linux, Windows, Mac

NVIDIA® Nsight™ Visual Studio Edition - CUDA, OpenGL and system trace and CUDA kernel profiling
  Integration into Visual Studio 2012 and newer - Windows only
NSIGHT SYSTEMS

Features

Multicore CPU and multi-GPU system activity trace

CPU utilization, thread state and core/thread migration

OS library trace with blocking call backtraces
  - pthread, semaphore, file I/O, network I/O, poll/select, sleep/yield, ioctl, syscall, fork

CUDA, OpenACC, OpenGL API and GPU Workload trace
  Includes UVM events
  cuDNN and cuBLAS

NVidia Tools eXtension (NVTX) user annotations

Command line