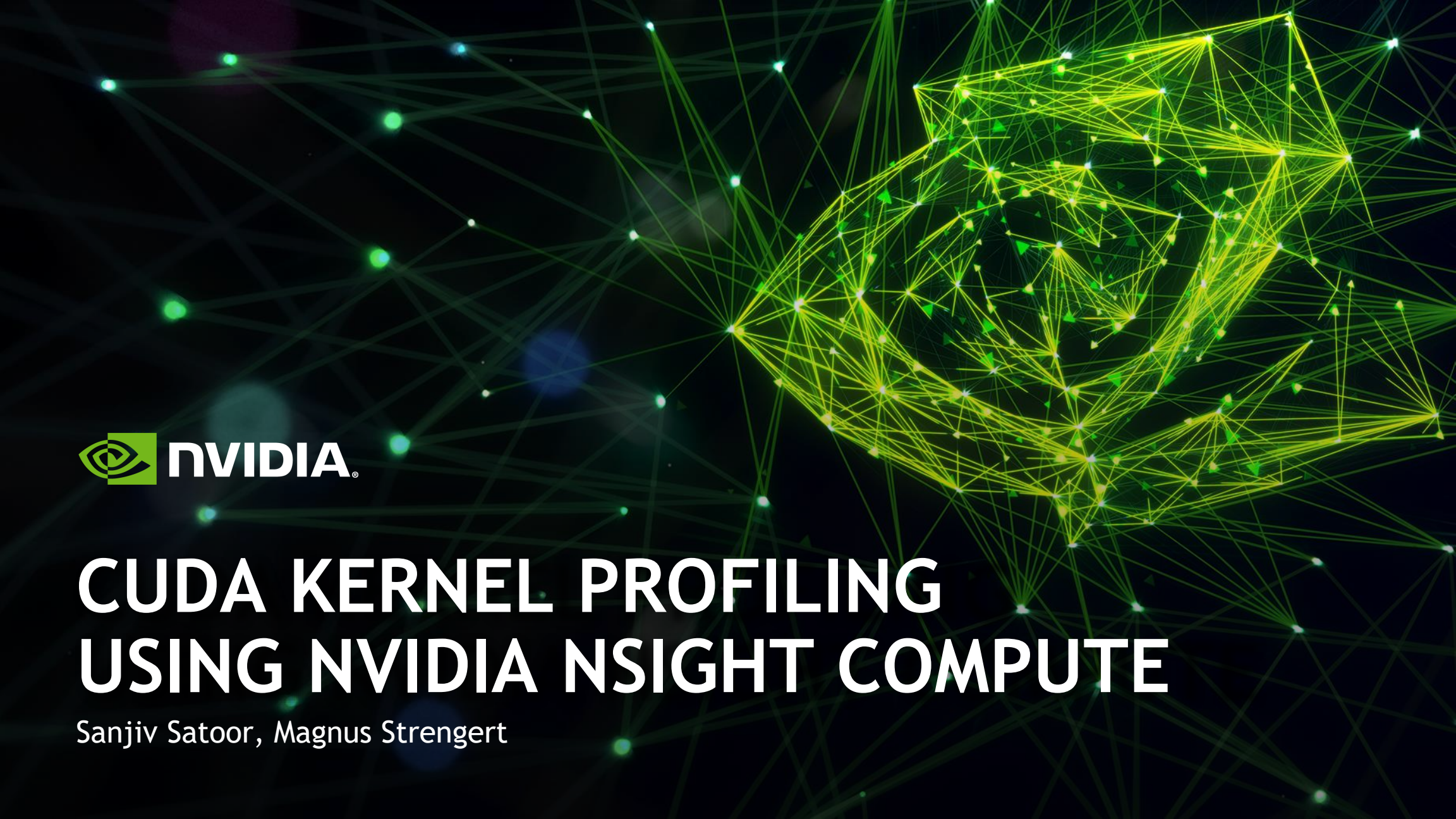


CUDA KERNEL PROFILING USING NVIDIA NSIGHT COMPUTE

Sanjiv Satoor, Magnus Strengert

AGENDA

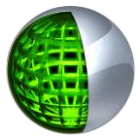
Overview of NVIDIA's Devtools

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)

eclipse

Visual Studio

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

NSIGHT COMPUTE

Updates for CUDA 10.1

Updates:

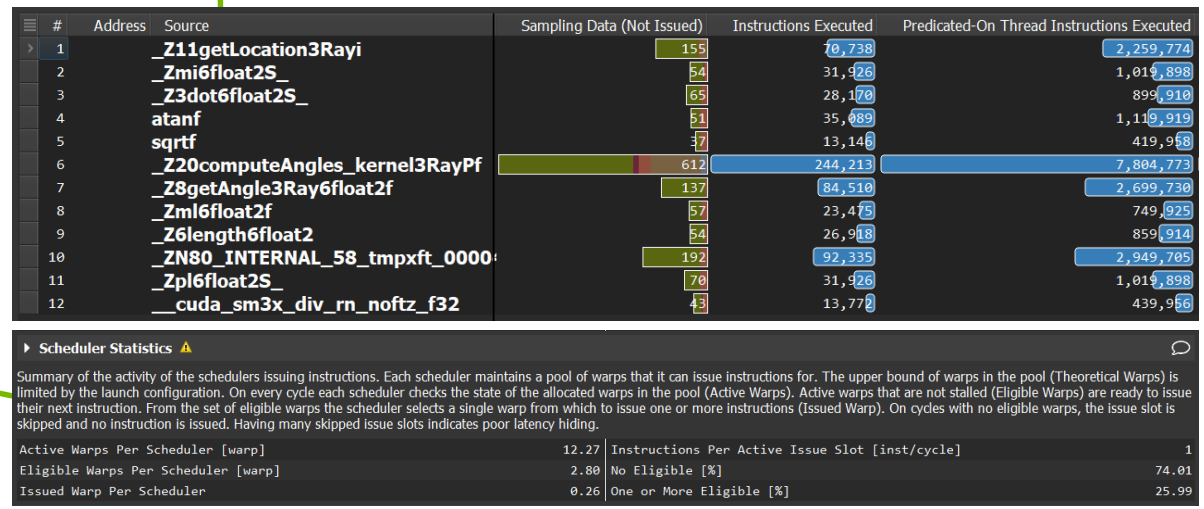
- Lower performance overhead and reduced memory overhead
- Source Page reports metrics by functions or files
- Updates sections and rules; section descriptions

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



#	Address	Source	Sampling Data (Not Issued)	Instructions Executed	Predicated-On Thread Instructions Executed
1		_Z11getLocation3Rayi	155	70,738	2,259,774
2		_Zmi6float2S_	54	31,926	1,019,898
3		_Z3dot6float2S_	65	28,170	899,910
4		atanf	51	35,089	1,119,919
5		sqrtf	17	13,148	419,958
6		_Z20computeAngles_kernel3RayPf	612	244,213	7,804,773
7		_Z8getAngle3Ray6float2f	137	84,510	2,699,730
8		_Zml6float2f	57	23,475	749,925
9		_Z6length6float2	54	26,918	859,914
10		_ZN80_INTERNAL_58_tmpxft_0000	192	92,335	2,949,705
11		_Zpl6float2S_	70	31,926	1,019,898
12		__cuda_sm3x_div_rn_noftz_f32	43	13,772	439,956

Scheduler Statistics

Summary of the activity of the schedulers issuing instructions. Each scheduler maintains a pool of warps that it can issue instructions for. The upper bound of warps in the pool (Theoretical Warps) is limited by the launch configuration. On every cycle each scheduler checks the state of the allocated warps in the pool (Active Warps). Active warps that are not stalled (Eligible Warps) are ready to issue their next instruction. From the set of eligible warps the scheduler selects a single warp from which to issue one or more instructions (Issued Warp). On cycles with no eligible warps, the issue slot is skipped and no instruction is issued. Having many skipped issue slots indicates poor latency hiding.

Active Warps Per Scheduler [warp]	12.27	Instructions Per Active Issue Slot [inst/cycle]	1
Eligible Warps Per Scheduler [warp]	2.80	No Eligible [%]	74.01
Issued Warp Per Scheduler	0.26	One or More Eligible [%]	25.99

NSIGHT PRODUCT FAMILY

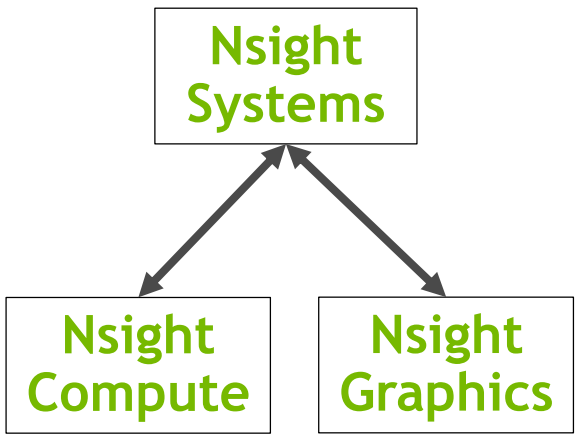
Standalone Performance Tools

Nsight Systems - System-wide application algorithm tuning

Nsight Compute - Debug/optimize specific CUDA kernel

Nsight Graphics - Debug/optimize specific graphics shader

Workflow



IDE Plugins

Nsight Eclipse Edition/Visual Studio – editor, debugger, some perf analysis

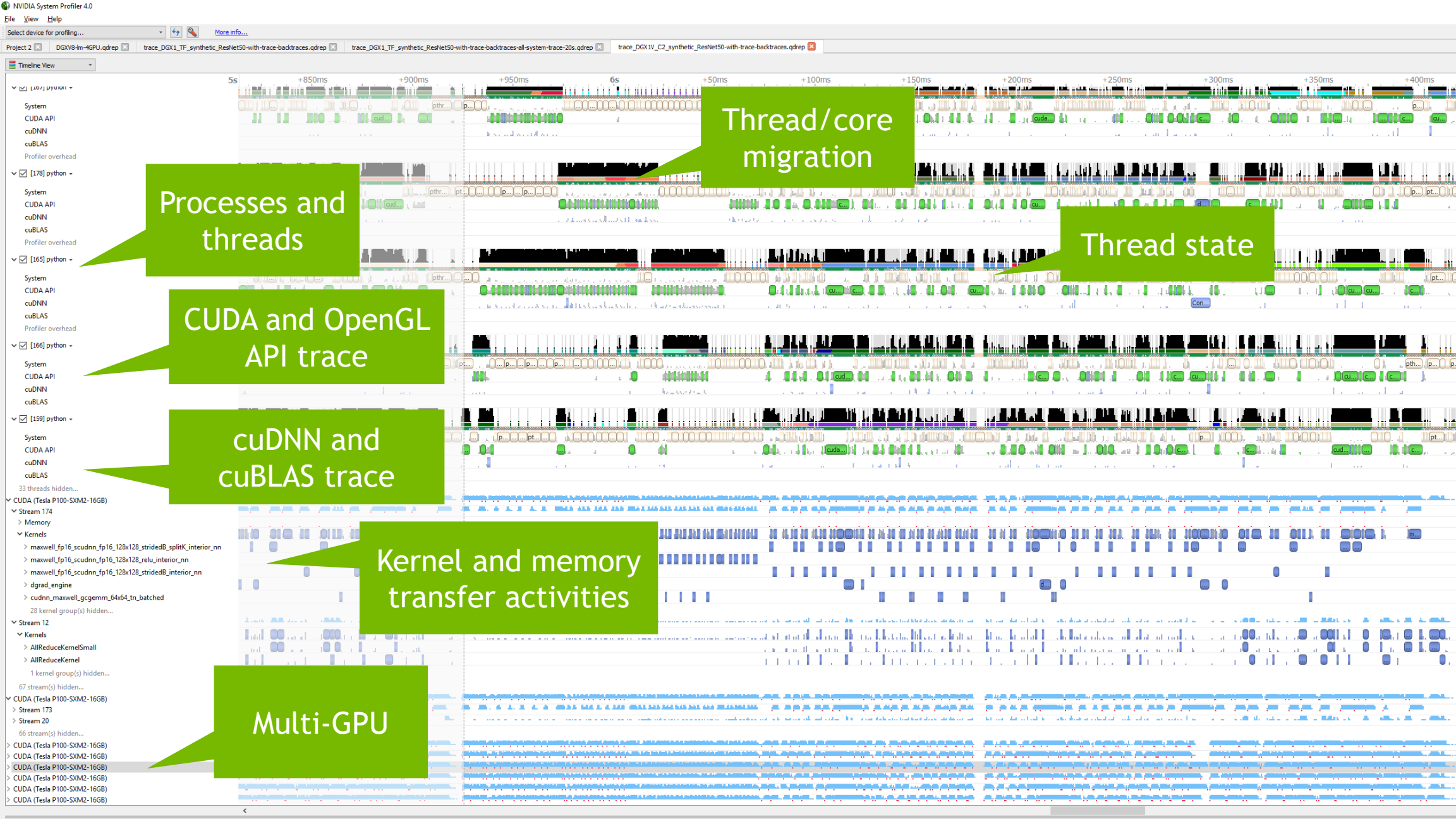
NSIGHT SYSTEMS

Overview

System-wide application algorithm tuning
- Multi-process tree support

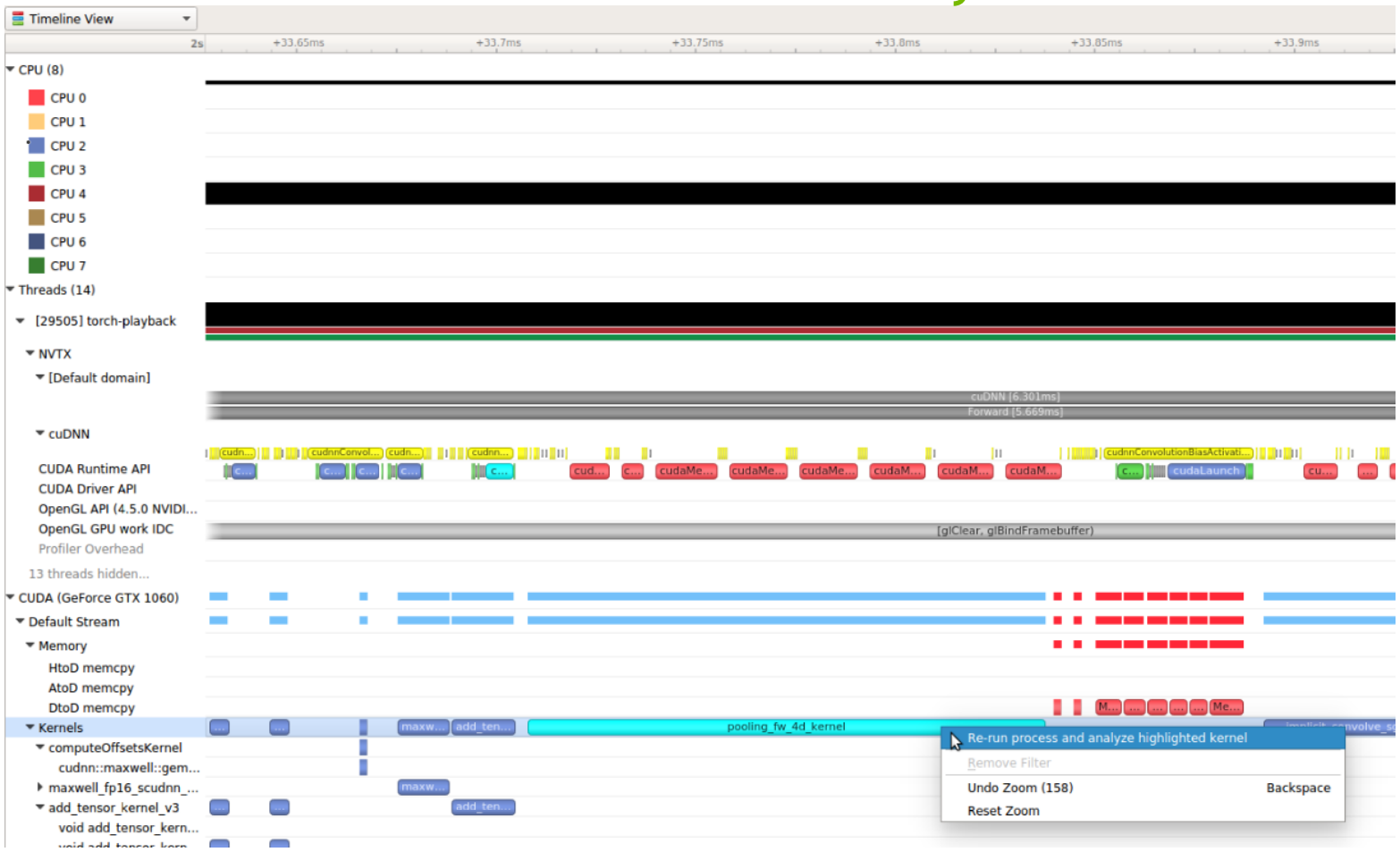
Locate optimization opportunities
- Visualize millions of events on a very fast GUI timeline
- Or gaps of unused CPU and GPU time

Balance your workload across multiple CPUs and GPUs
- CPU algorithms, utilization, and thread state
- GPU streams, kernels, memory transfers, etc


NVIDIA System Profiler 4.0
File View Help
Select device for profiling...
More info...
Project 2
DGXV8-lm-4GPU.qdrep
trace_DGX1_TF_synthetic_ResNet50-with-trace-backtraces.qdrep
trace_DGX1_TF_synthetic_ResNet50-with-trace-backtraces-all-system-trace-20s.qdrep
trace_DGX1V_C2_synthetic_ResNet50-with-trace-backtraces.qdrep
Timeline View
5s
+850ms
+900ms
+950ms
6s
+50ms
+100ms
+150ms
+200ms
+250ms
+300ms
+350ms
+400ms
System
CUDA API
cuDNN
cuBLAS
Profiler overhead
[178] python
[165] python
[166] python
[159] python
33 threads hidden...
CUDA (Tesla P100-SXM2-16GB)
Stream 174
Memory
Kernels
maxwell_fp16_scudnn_fp16_128x128_stridedB_splitK_interior_nn
maxwell_fp16_scudnn_fp16_128x128_relu_interior_nn
maxwell_fp16_scudnn_fp16_128x128_stridedB_interior_nn
dgrad_engine
cudnn_maxwell_gcgemm_64x64_tn_batched
28 kernel group(s) hidden...
Stream 12
AllReduceKernelSmall
AllReduceKernel
1 kernel group(s) hidden...
67 stream(s) hidden...
Stream 173
Stream 20
66 stream(s) hidden...
Thread/core migration
Processes and threads
Thread state
CUDA and OpenGL API trace
cuDNN and cuBLAS trace
Kernel and memory transfer activities
Multi-GPU

TRANSITIONING TO PROFILE A KERNEL

Dive into kernel analysis



NSIGHT COMPUTE INTRODUCTION

NVIDIA NSIGHT COMPUTE

Next-Gen Kernel Profiling Tool

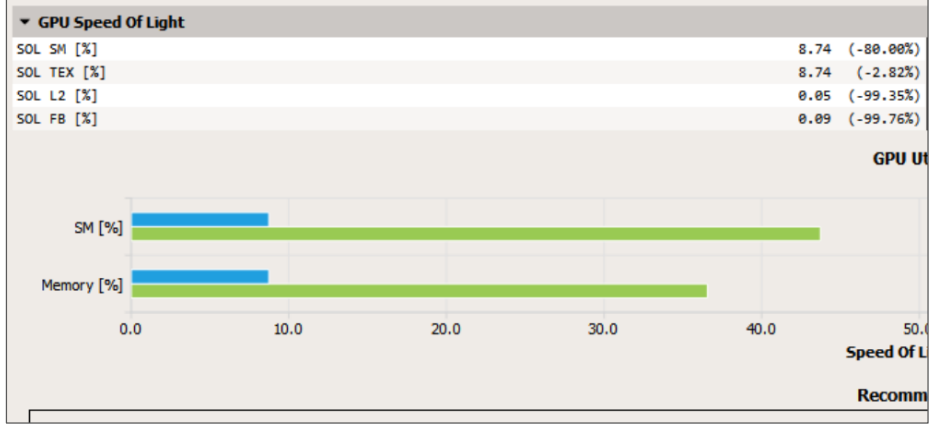
Key Features:

- 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

Docs/product: https://developer.nvidia.com/nsight-compute



inst_executed [inst]	16,528.00; 16,528.00; …	13,476.00; 13,476.00; …
l1tex__sol_pct [%]	14.33	n/a
launch__block_size	128.00	128.00
launch__function_pcs	47,611,587,968.00	12,273,728.00
launch__grid_size	4,132.00	3,369.00
launch__occupancy_limit_blocks [block]	32.00	32.00
launch__occupancy_limit_registers [register]	21.00	21.00
launch__occupancy_limit_shared_mem [bytes]	384.00	384.00
launch__occupancy_limit_warps [warps]	16.00	16.00
launch__occupancy_per_block_size	3,638.00	3,638.00
launch__occupancy_per_register_count	5,792.00	5,792.00
launch__occupancy_per_shared_mem_size	2,260.00	2,260.00
launch__registers_per_thread [register/thread]	17.00	17.00
launch__shared_mem_config_size [bytes]	49,152.00	49,152.00
launch__shared_mem_per_block_dynamic [bytes/block]	0.00	0.00
launch__shared_mem_per_block_static [bytes/block]	20.00	20.00
launch__thread_count [thread]	528,896.00	431,232.00
launch__waves_per_multiprocessor	3.23	42.11
ltc__sol_pct [%]	6.93	7.18
memory_access_size_type [bytes]	2.00; 32.00; 32.00; 32…	2.00; 32.00; 32.00; 32…

Source	Live Registers	Sampling Data (All)	Sampling Data (No Issue)
@!PT SHFL.IDX PT, RZ, RZ, RZ, RZ;	0	223	0
MOV R1, c[0x0][0x28];	1	13	44
S2R R0, SR_CTAID.X;	2	143	75
S2R R2, SR_TID.X;	3	0	38
IMAD R0, R0, c[0x0][0x0], R2;	3	599	94
ISETP.GE.AND P0, PT, R0, c[0x0][0x170]	2	125	26
@P0 EXIT;	2	259	86
MOV R2, R0;	3	386	29
@!PT SHFL.IDX PT, RZ, RZ, RZ, RZ;	2	0	0
MOV R4, 0x4;	3	0	0
IMAD.WIDE R4, R2, R4, c[0x0][0x160];	4	0	0
LDG.E.SYS R3, [R4];	3	0	0
BSSY B0, 0xb00976780;	3	0	0
SHF.R.S32.HI R0, RZ, 0x1f, R2;	4	0	0


NSIGHT COMPUTE
DEMO

WARP SCHEDULER

Volta Architecture

4 Warp Scheduler per SM

Manages a pool of warps:

Volta: 16 warp slots

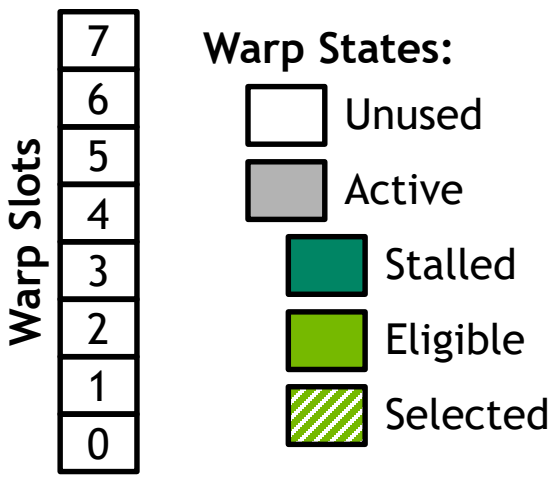
Turing: 8 warp slots

Each scheduler can issue 1 warp/cycle

Offers simplified mental model for profiling and SM metrics

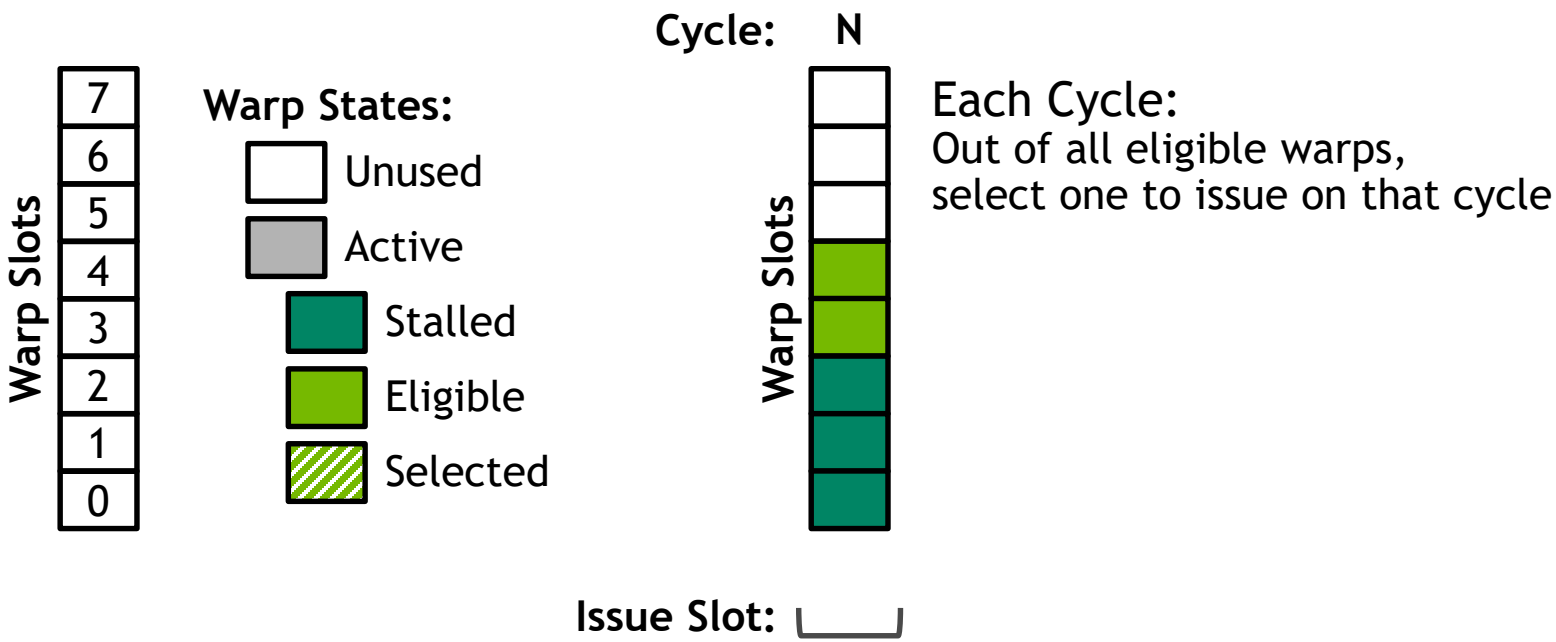
WARP SCHEDULER

Mental Model for Profiling



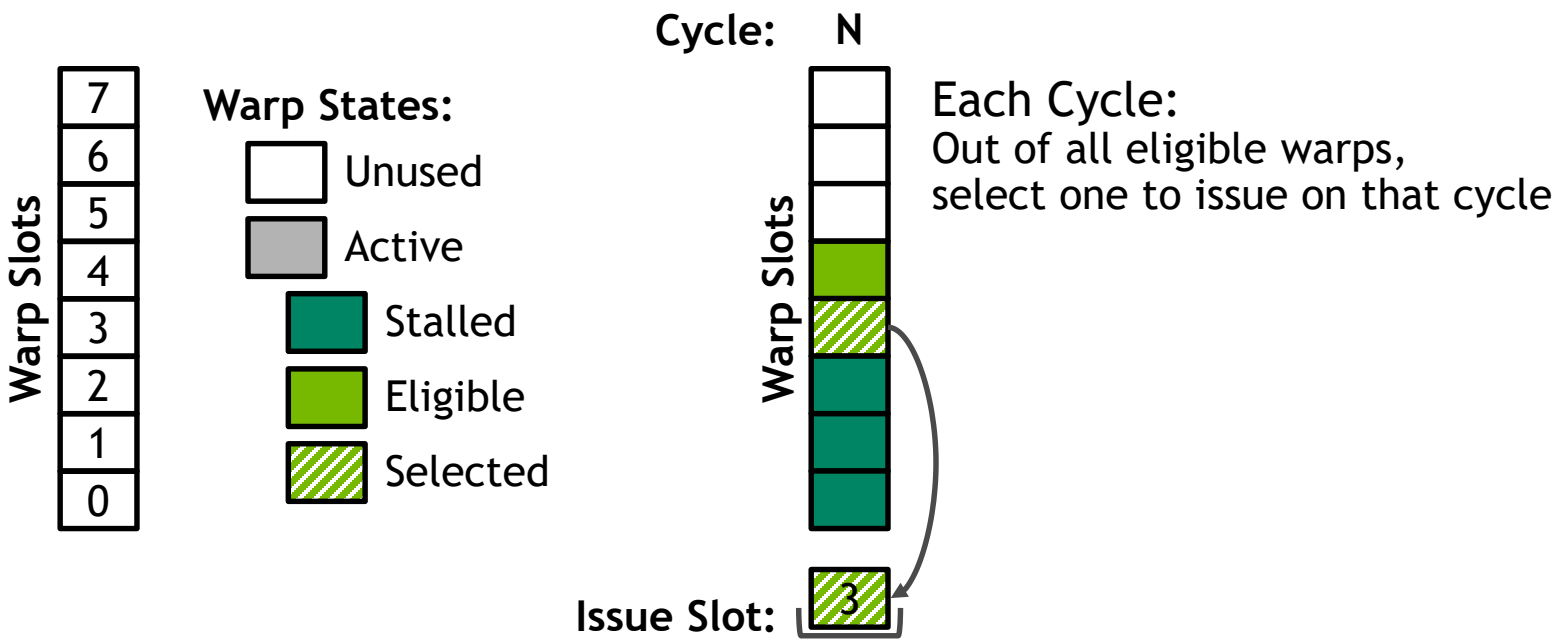
WARP SCHEDULER

Mental Model for Profiling



WARP SCHEDULER

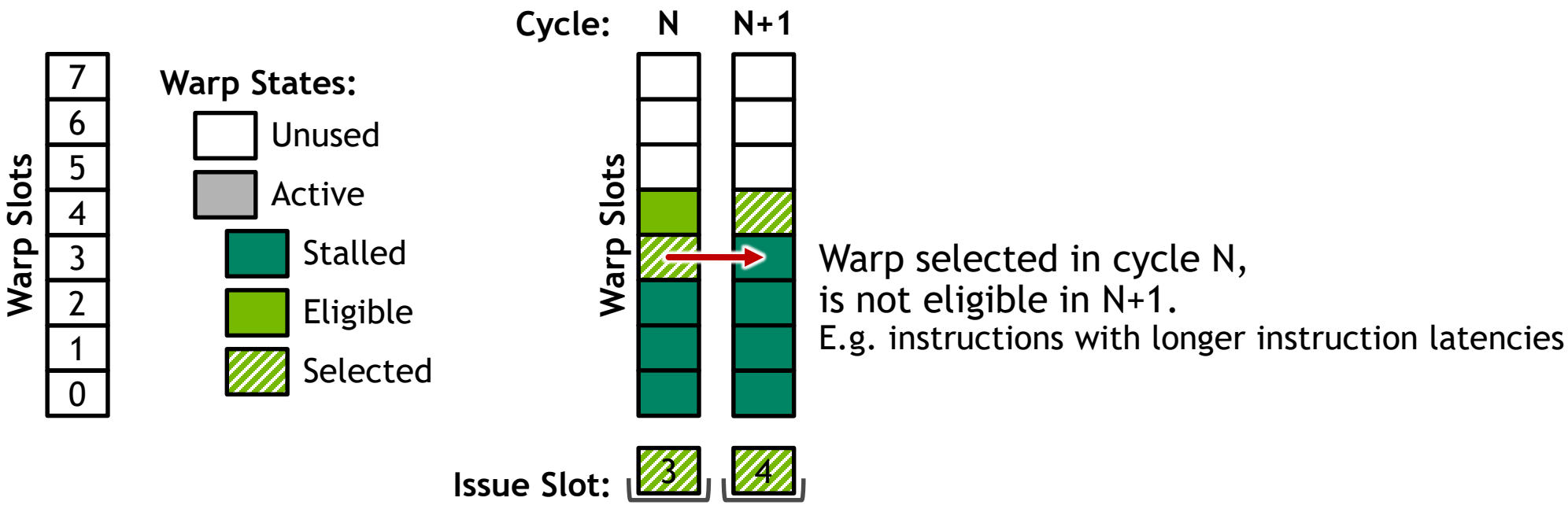
Mental Model for Profiling



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

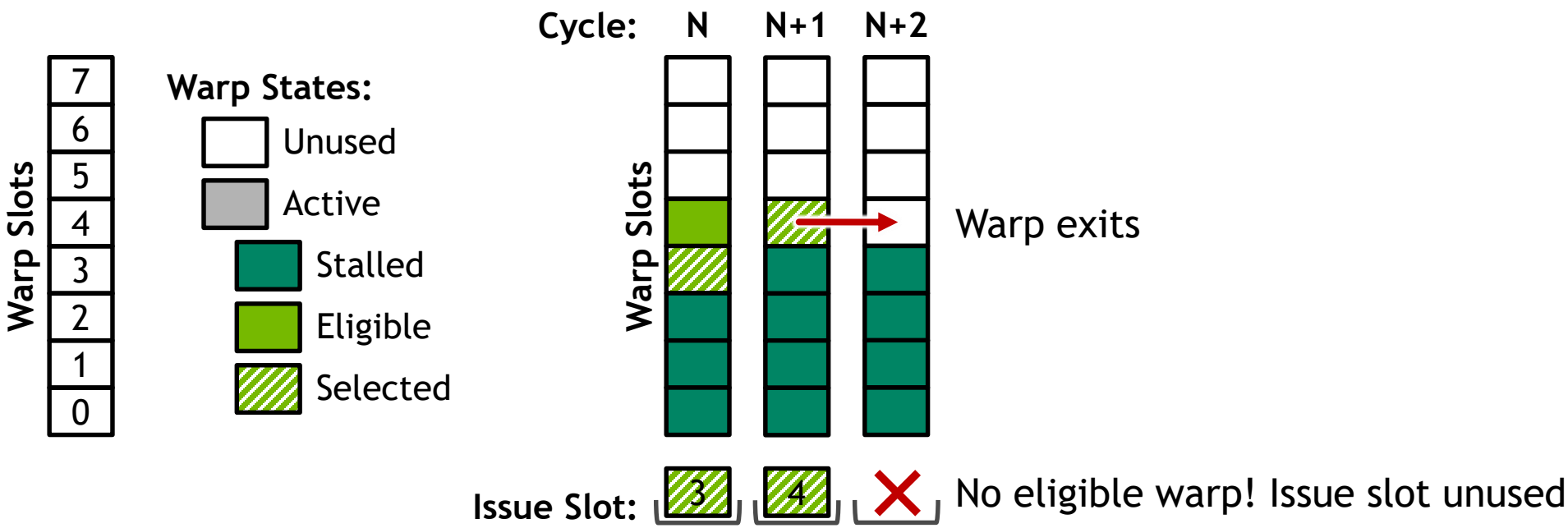
WARP SCHEDULER

Mental Model for Profiling



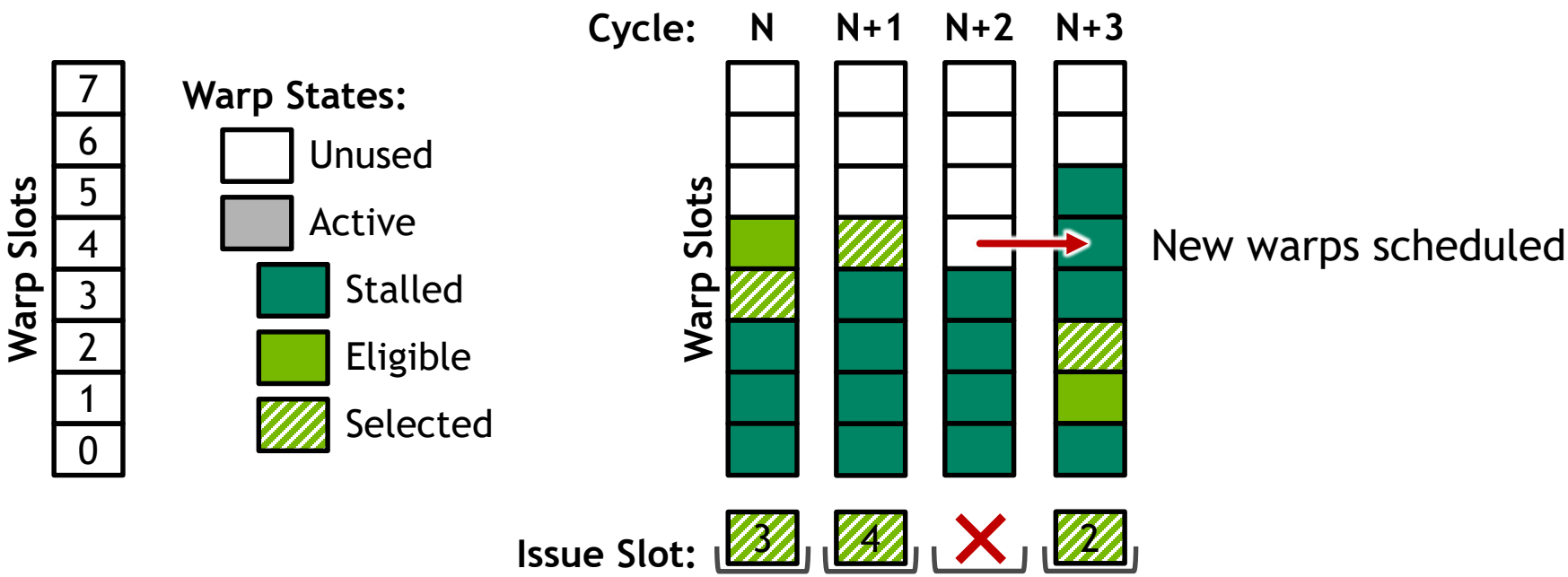
WARP SCHEDULER

Mental Model for Profiling



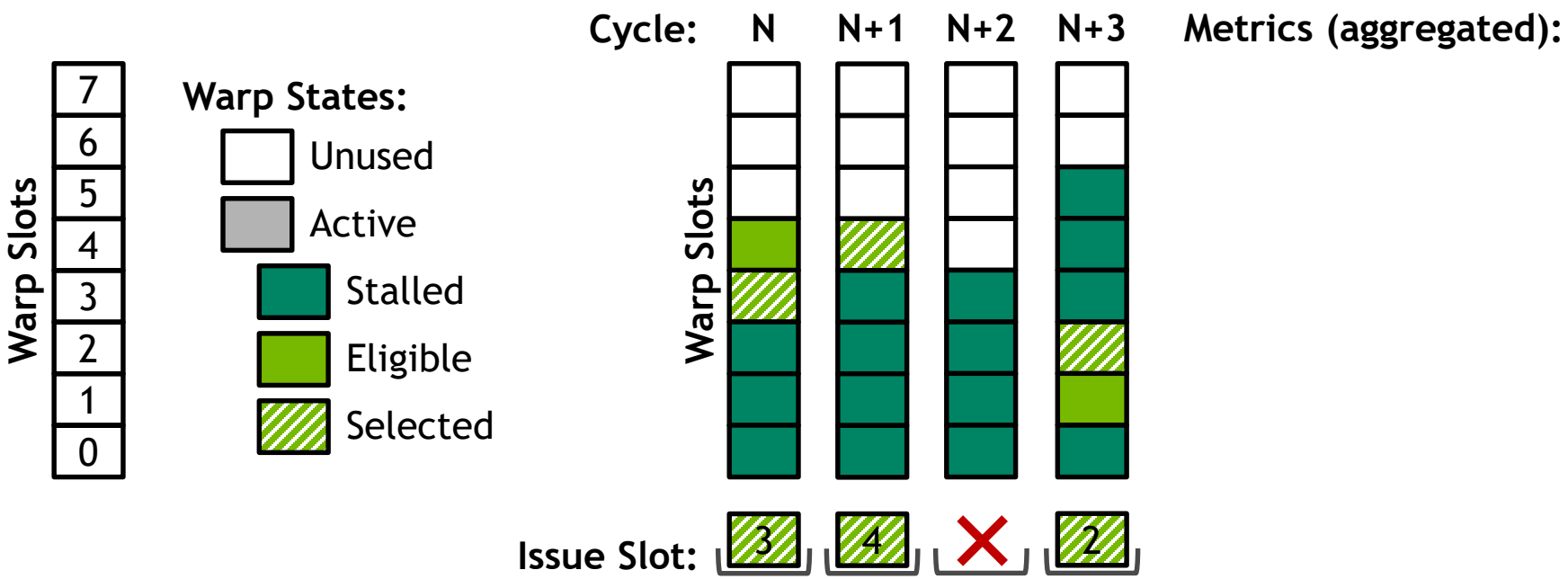
WARP SCHEDULER

Mental Model for Profiling



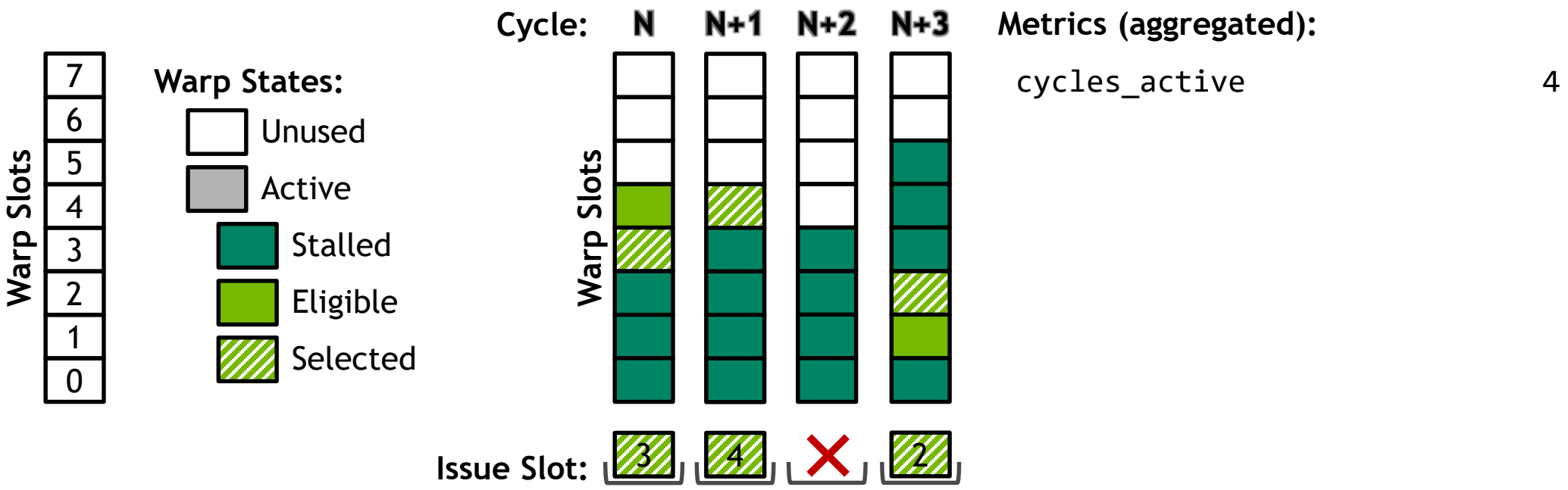
WARP SCHEDULER

Mental Model for Profiling



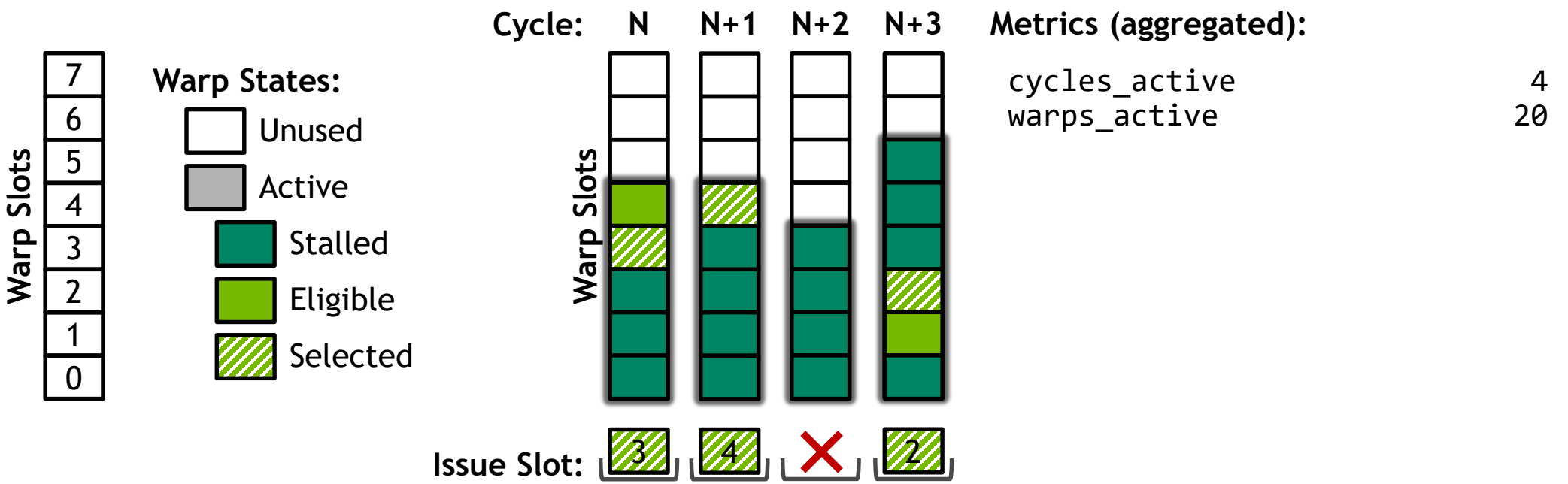
WARP SCHEDULER

Mental Model for Profiling



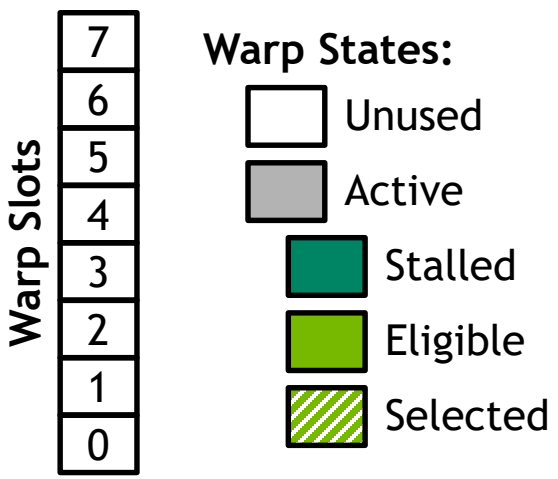
WARP SCHEDULER

Mental Model for Profiling



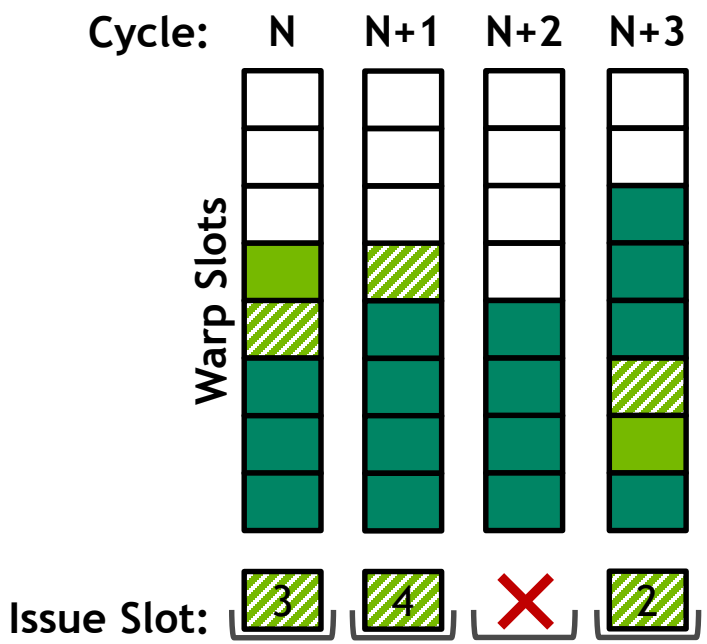
WARP SCHEDULER

Mental Model for Profiling



Warp States:

- Unused
- Active
- Stalled
- Eligible
- Selected

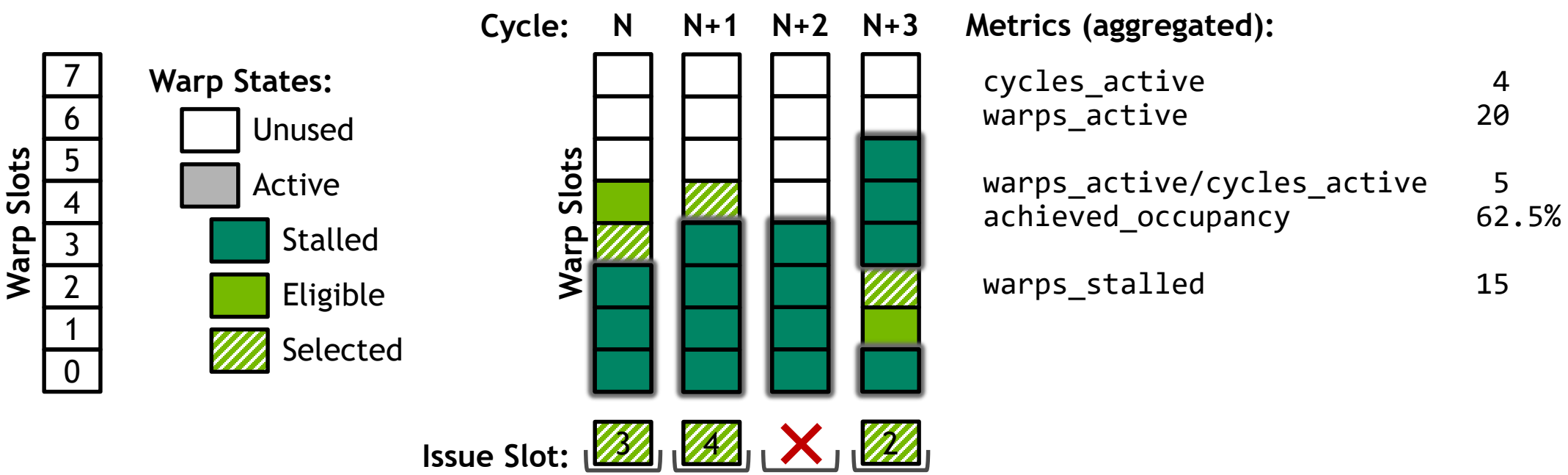


Metrics (aggregated):

cycles_active	4
warps_active	20
warps_active/cycles_active	5
achieved_occupancy	62.5%

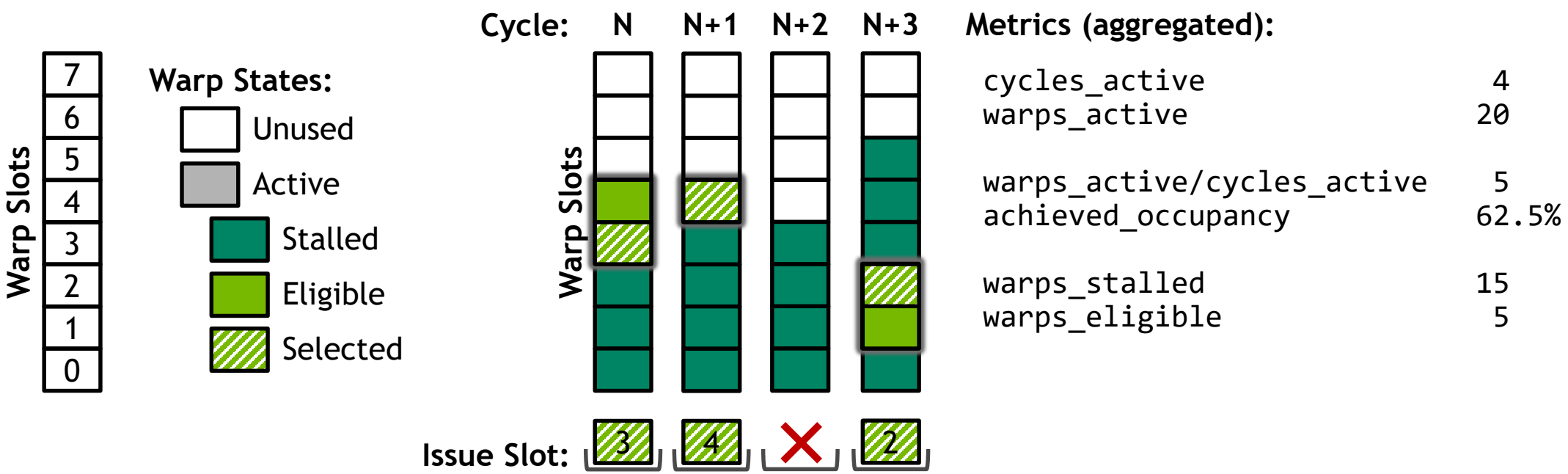
WARP SCHEDULER

Mental Model for Profiling



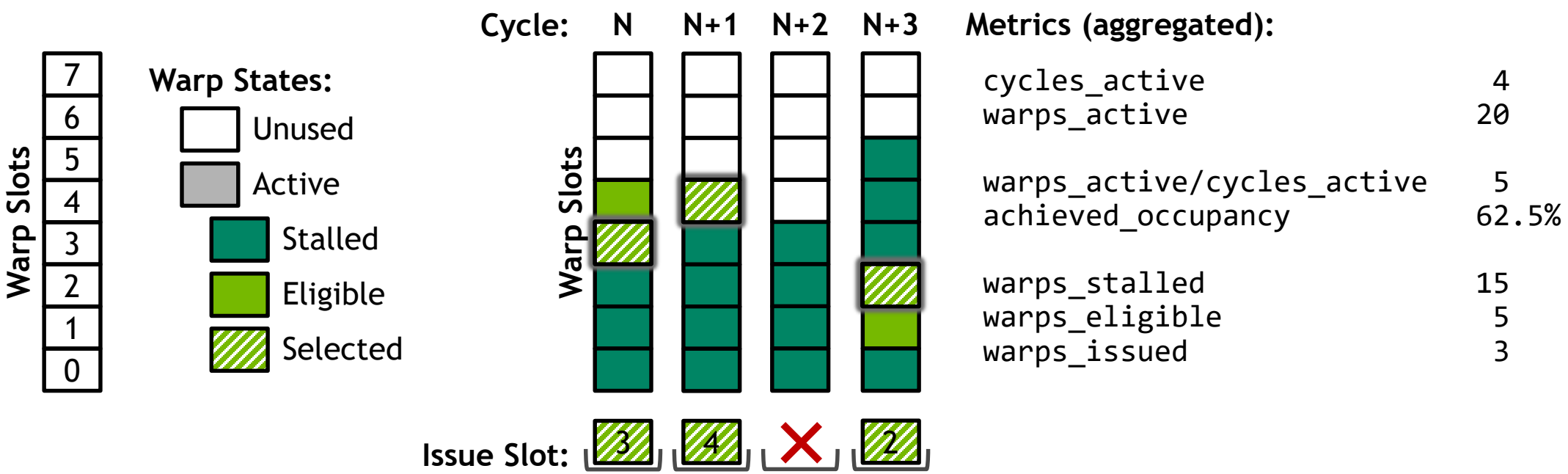
WARP SCHEDULER

Mental Model for Profiling



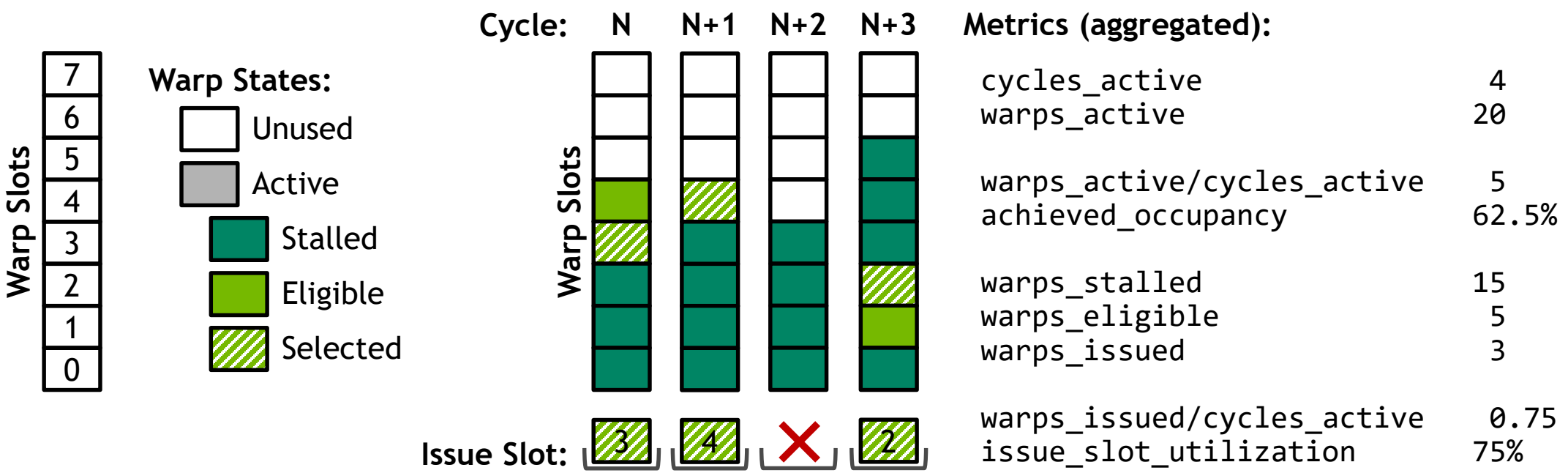
WARP SCHEDULER

Mental Model for Profiling



WARP SCHEDULER

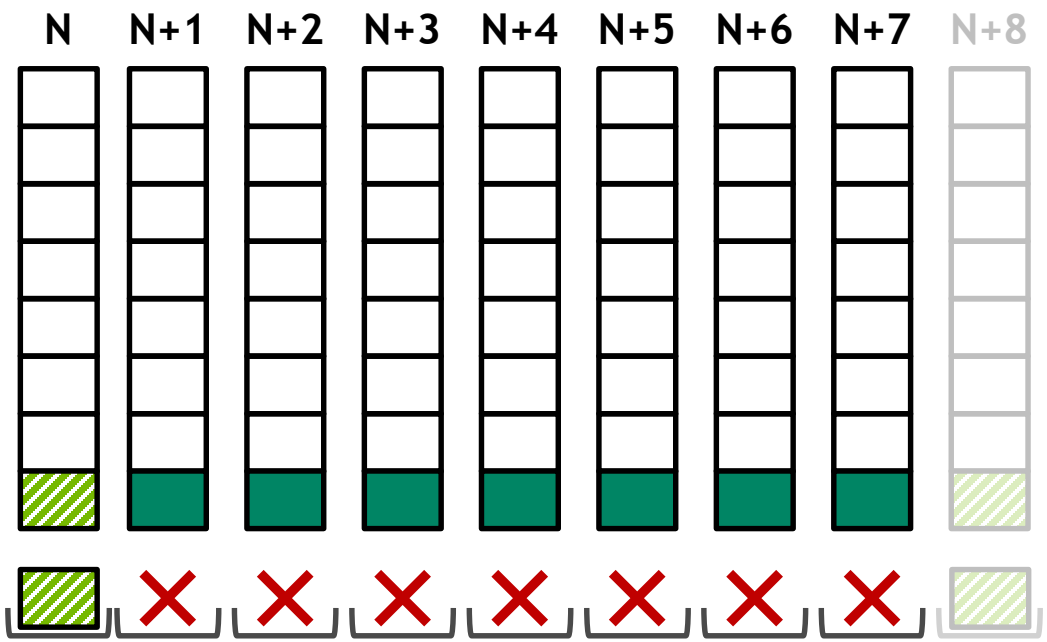
Mental Model for Profiling



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

WARP SCHEDULER

Application to kernel_B



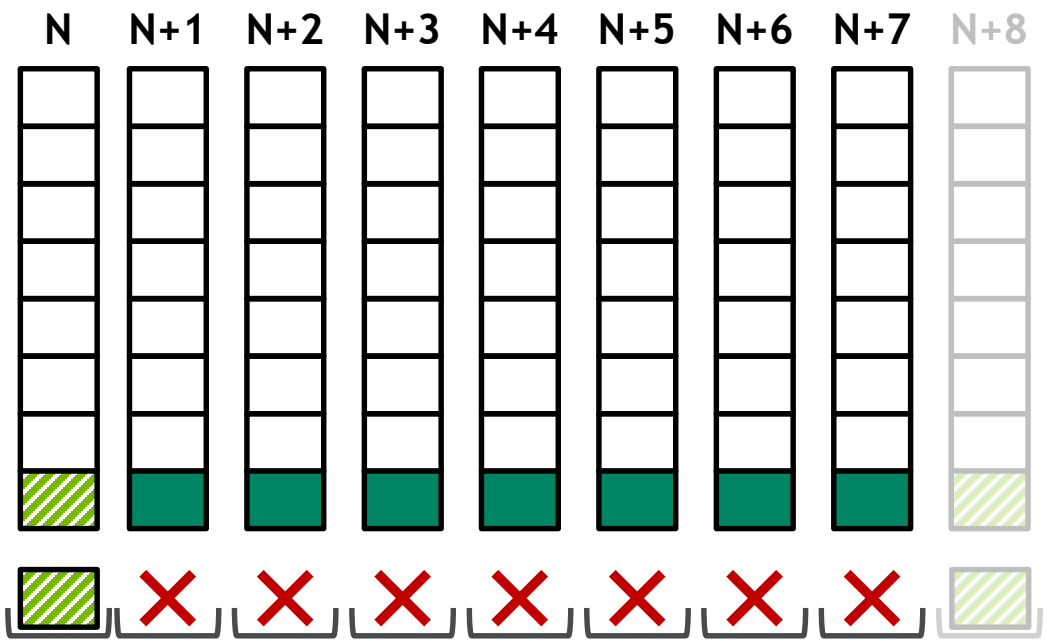
Metrics (theoretical; every 8 cycles):

warps_active	8	1
warps_stalled	7	7/8
warps_eligible	1	1/8
warps_selected	1	1/8

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

WARP SCHEDULER

Application to kernel_B



Metrics (theoretical; every 8 cycles):

warps_active	8	1
warps_stalled	7	7/8
warps_eligible	1	1/8
warps_selected	1	1/8

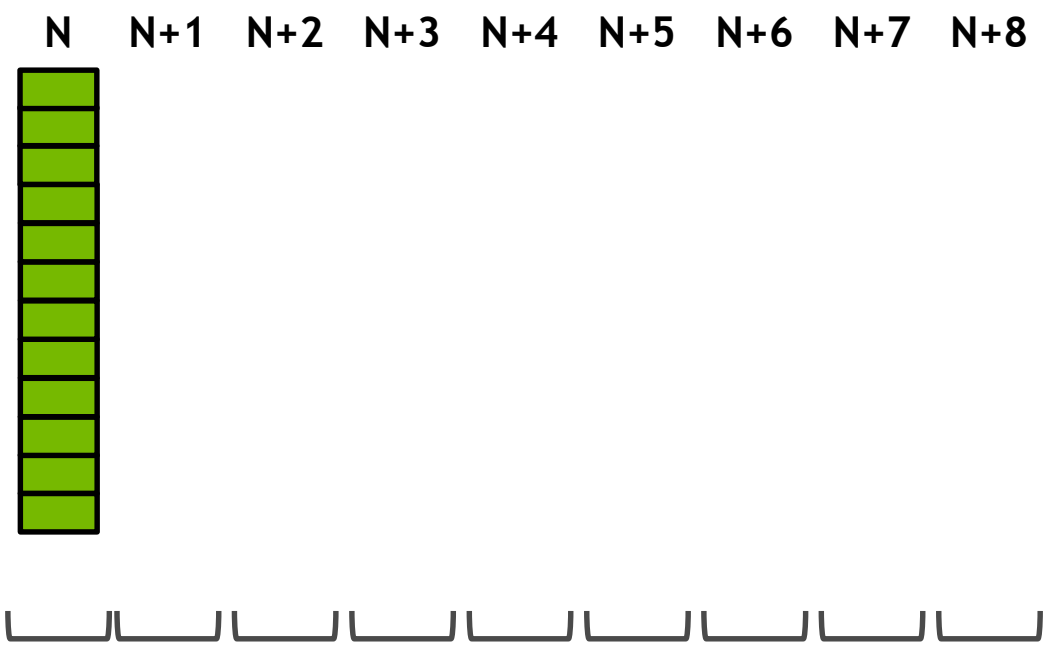
Metrics (from report):

warps_active	1.00
warps_stalled	0.87
warps_eligible	0.13
warps_selected	0.13

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

WARP SCHEDULER

Application to kernel_A



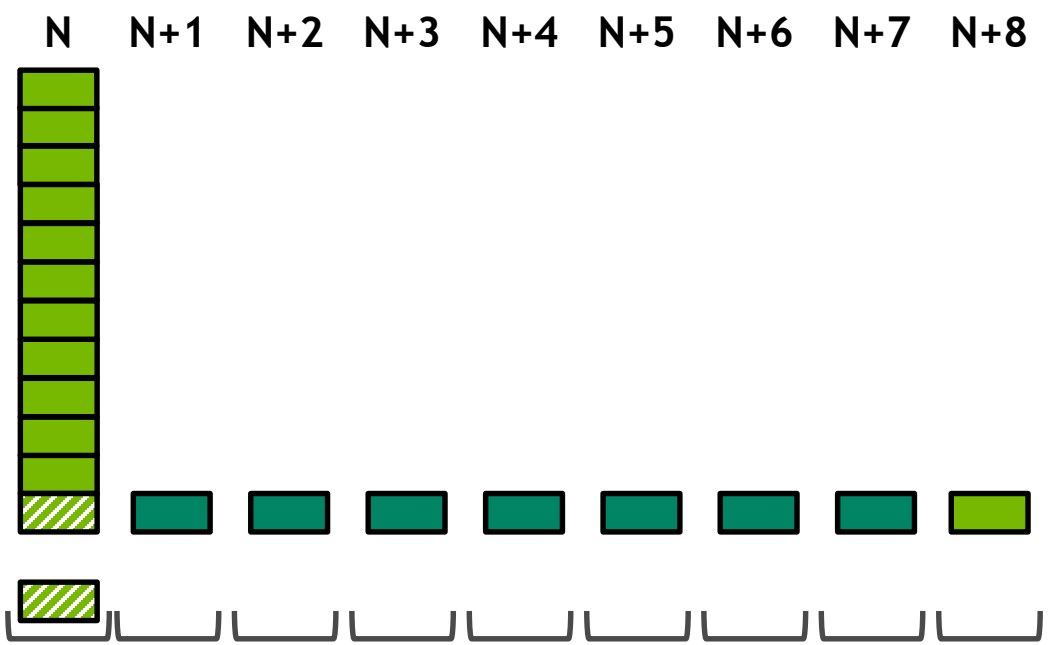
FP64 Pipeline on Volta:

Dependent Issue Rate: 8 cycles
Issue Rate: 4 cycles

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

WARP SCHEDULER

Application to kernel_A



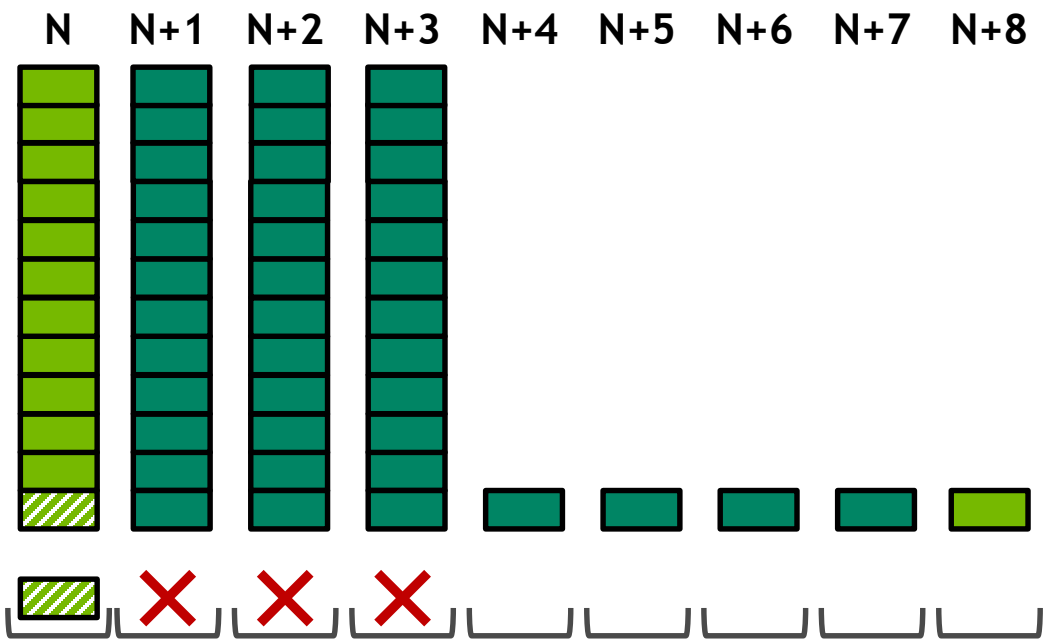
FP64 Pipeline on Volta:

Dependent Issue Rate: 8 cycles
Issue Rate: 4 cycles

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

WARP SCHEDULER

Application to kernel_A



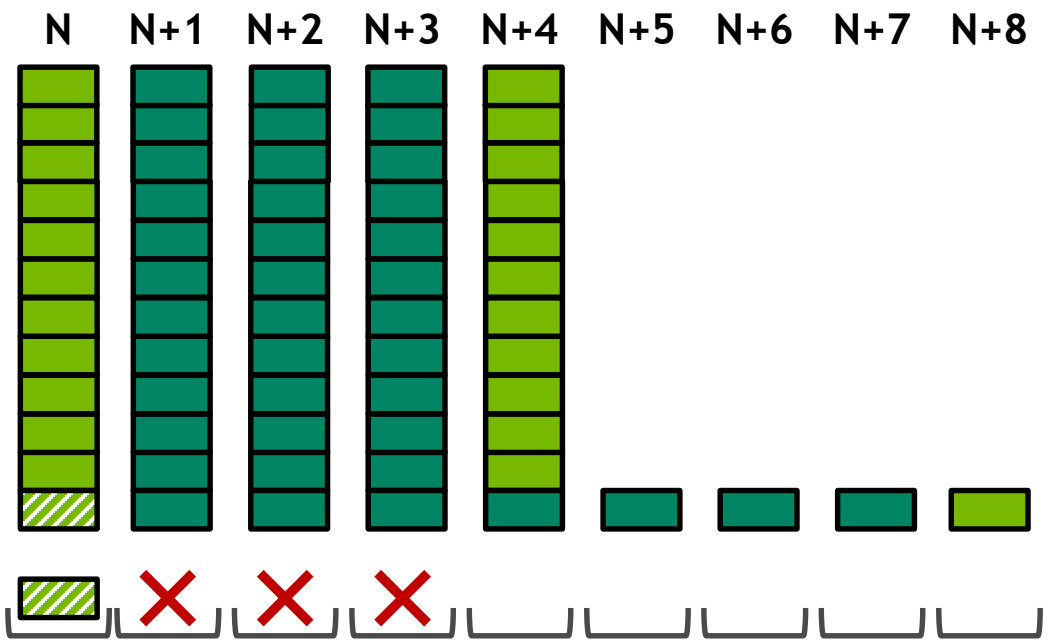
FP64 Pipeline on Volta:

Dependent Issue Rate: 8 cycles
Issue Rate: 4 cycles

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

WARP SCHEDULER

Application to kernel_A



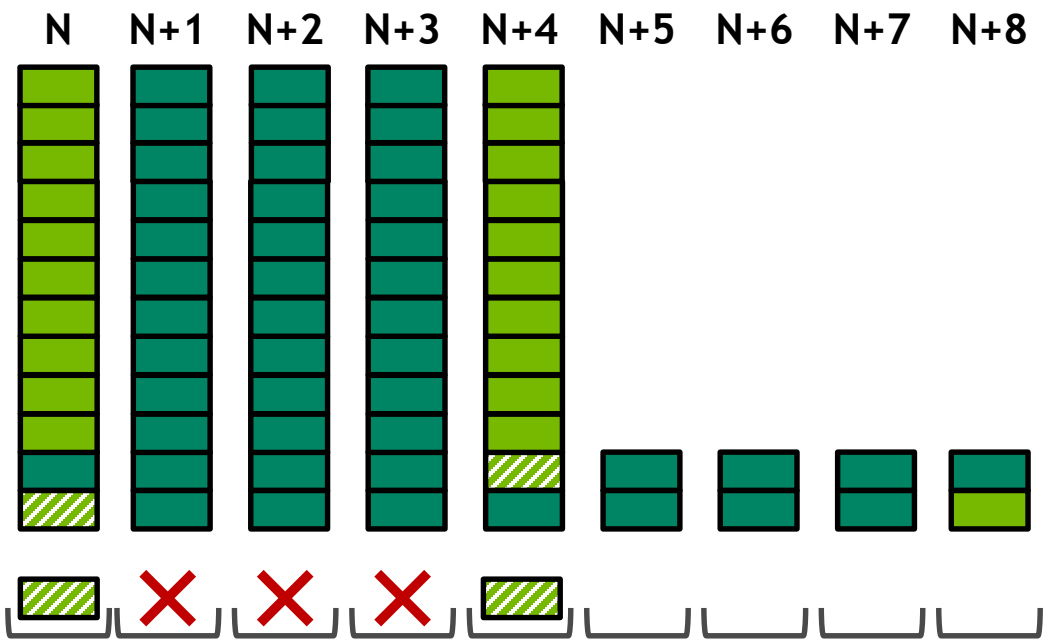
FP64 Pipeline on Volta:

Dependent Issue Rate: 8 cycles
Issue Rate: 4 cycles

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

WARP SCHEDULER

Application to kernel_A



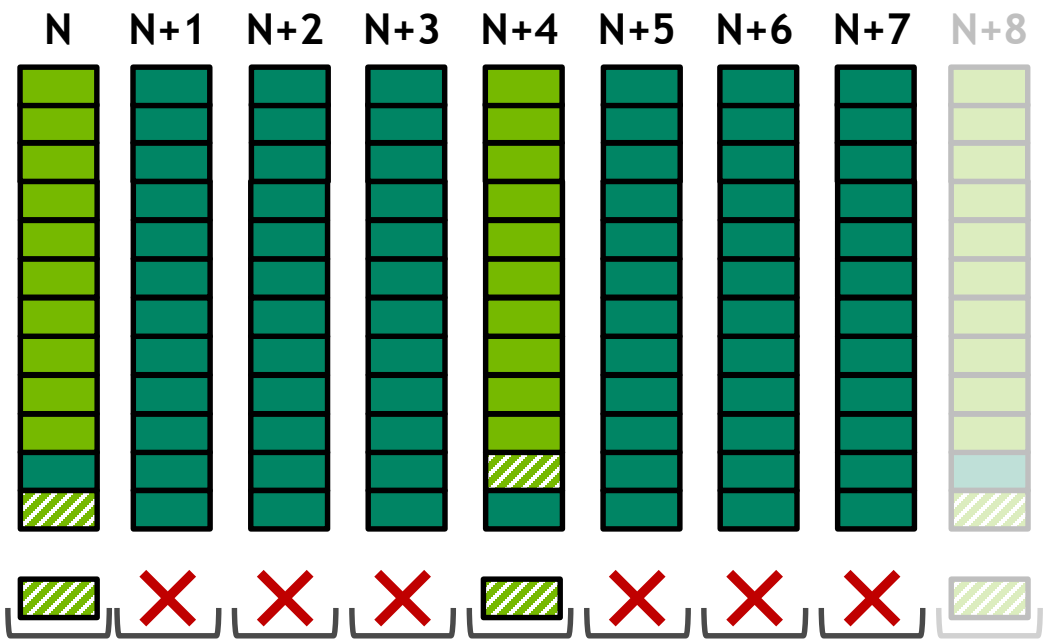
FP64 Pipeline on Volta:

Dependent Issue Rate: 8 cycles
Issue Rate: 4 cycles

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

WARP SCHEDULER

Application to kernel_A



Metrics (theoretical; every 8 cycles):

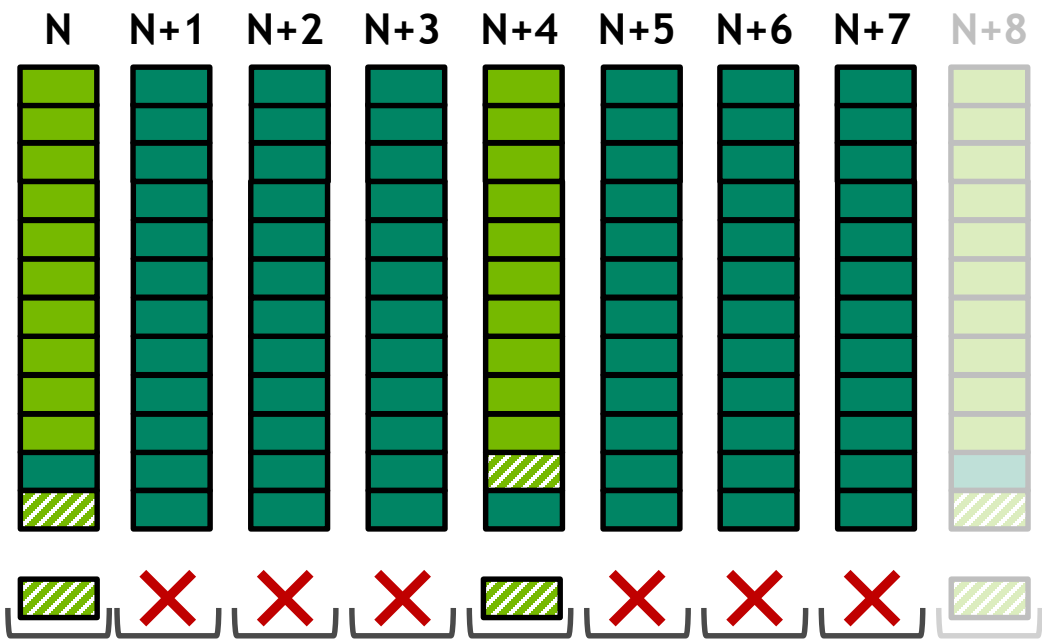
warps_active	96
warps_stalled	74
warps_eligible	22
warps_selected	2

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


THANK YOU
Any Questions?
NVIDIA

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