CONNECTION INSTRUCTIONS

- Navigate to nvlabs.qwiklab.com
- Login or create a new account
- Select the "Instructor-Led Hands-on Labs" class
- Find the lab called "Optimizing CUDA Application Performance..." and click Start
- After a short wait, lab instance connection information will be shown
- Please ask Lab Assistants for help!

GPU TECHNOLOGY CONFERENCE

OPTIMIZING CUDA APPLICATION PERFORMANCE WITH NVIDIA'S VISUAL PROFILER

YU ZHOU (NVIDIA)

MAYANK KAUSHIK (NVIDIA)

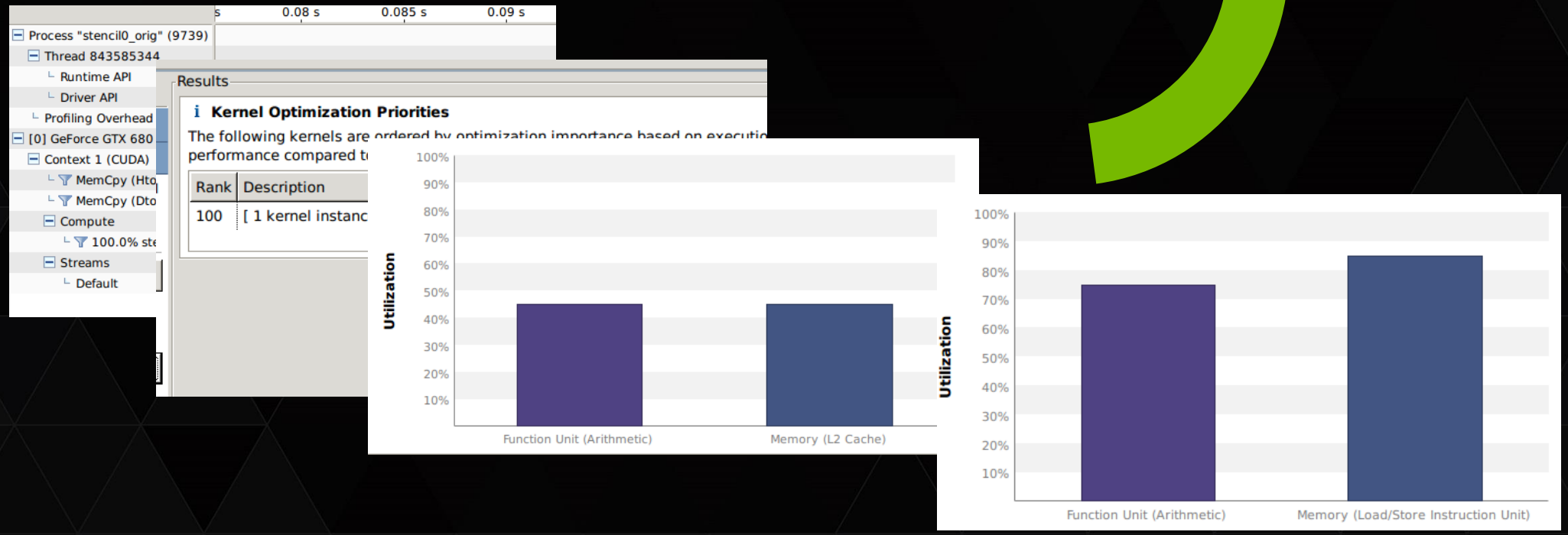
```
// Executes for each pixel
__global__ void stencil_kernel(...) {

  ...

  foreach (adjacent pixels) {
    foreach (color channels) {
      out[index] += in[index + radius, channel] * weight[radius];
    }
  }
}

cudaMemcpy(..., in, SIZE, H2D);
stencil_kernel<<< ceil(#pixels/BLOCK_SIZE), BLOCK_SIZE >>>(...);
cudaMemcpy(out, ..., SIZE, D2H);
```

HOW TO COMPILE/RUN

- **cd ~/gtc2015**
- **make stepX** (X=0,1,..,5) → stencilX_* executable
- Modify “stencilX_*.cu”
- **make clean** to restore
- ~/gtc2015/instructions.pdf

HOW TO PROFILE

- **Visual Profiler** shortcut on Desktop
- Iterative approach



STEP0

- cd ~/gtc2015
- make step0
- (in Visual Profiler) “File” -> “New Session”
- “Browse...” -> pick “stencil0_orig”
- “Next” -> “Finish”

STEP1

- “Examine Individual Kernels”
- Select kernel from the list
- Occupancy is low (note the “properties” window on the right)
- “Perform Kernel Analysis”
- “Perform Latency Analysis”
- Occupancy is limited by block size
- “make step1” (automatically patch the source to increase block size)
- Run stencil1_occu with the profiler
- Try to find how to improve performance further

STEP2

- “Examine Individual Kernels”
- Select kernel from the list
- Occupancy is above 90% now
- “Perform Kernel Analysis”
- “Perform Memory Bandwidth Analysis”
- L2 cache traffic is high (due to duplicated data transfers)
- “make step2” (automatically patch the source to use shared memory)
- Run stencil2_shm with the profiler
- Try to find how to improve performance further

STEP3

- “Examine Individual Kernels”
- Select kernel from the list
- “Perform Kernel Analysis”
- “Perform Memory Bandwidth Analysis” (still memory bound)
- L2 cache traffic is lower now
- But global memory access pattern is bad
- Click on the line/file location to jump to the problem in source file
- “make step3” (automatically patch the source to coalesce memory transfer)
- Run stencil3_coalesce with the profiler
- Try to find how to improve performance further

STEP4

- “Examine Individual Kernels”
- Select kernel from the list
- “Perform Kernel Analysis”
- “Perform Compute Analysis” (the kernel became compute-bound)
- The “Integer” function unit utilization is high
- “Show Kernel Profile”
- Click on the name of the kernel to see what the hot spot is
- “make step4” (automatically patch the source to distribute computation over integer/fp units)
- Run stencil4_fp with the profiler
- Try to find how to improve performance further

STEP5

- Notice now kernel only occupies a small portion of the whole timeline
- Observe low overlap between activities on the timeline
- Check GPU properties; it is capable of doing 2 memcpys at the same time
- “make step5” (automatically patch the source to divide data into chunks and pipeline the kernel)
- Observe on the new timeline that memcpys and kernels are now overlapped

WHAT'S NEXT?

- Download today!
  Search "**download cuda**"
  cudatools@nvidia.com

- S5174 - CUDA Optimization with NVIDIA Nsight Visual Studio Edition
  15:30 – 16:50, Room 210G

- S5655 - Hands-on Lab: CUDA Application Development Life Cycle
  Thur, 14:00 – 15:20, Room 211A

- Last year's sessions
  Search "GTC on demand"