GPUs for Science 2020

Tutorial: CUDA programming Session 3
Debugging

Tools to help debugging

printf:

- Is it really a tool? .. Basic but sometimes efficient enough
- Be aware: thousands of threads
  - if (threadIdx.x + blockIdx.x * blockDim.x == 0)

cuda-gdb:

- Similar to gdb (bt, info cuda threads, print, continue, step, n, quit)
- Choose context
  - cuda thread kernel,block,threaded

cuda-memcheck

- Suite of tools
  - Memcheck, racecheck, initcheck, synccheck

TotalView

- GUI debugger

Compiling flags

- Compiling with
  - -G to generate debug info, affect compiler optimizations
  - -lineinfo to generate line number information without affecting optimizations
  - -rdynamic for the host compiler to retain function symbols
  - –Xcompiler to specify flags to the host compiler
- Nvcc –lineinfo –Xcompiler –rdynamic –o bar.exe bar.cu

Cuda-gdb

- Set breakpoints (b myfunc.cu:48)
- Print values
  - P var
  - P *(array)@10
- Control execution
  - run, next, step, continue
- No watchpoints on CUDA (different from gdb)

- Change context

```
(cuda-gdb) cuda device sm warp lane            # display HW coords
device 0, sm 0, warp 0, lane 0
(cuda-gdb) cuda kernel block thread            # display SW coords
kernel 1, block (0,0,0), thread (0,0,0)
(cuda-gdb) cuda device 0 sm 1 warp 2 lane 3    # switch focus
[Switching focus to CUDA kernel 1, grid 2, block (8,0,0), thread (67,0,0), device 0, sm 1,
warp 2, lane 3]
374 int totalThreads = gridDim.x * blockDim.x
```

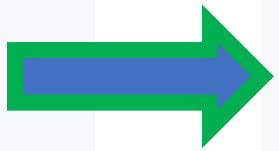
Cuda-gdb

- To execute
  - Inside interactive batch session
  - module load cuda
  - srun –pty cuda-gdb –args ./myprog.exe [arg1] [arg2] […]

Cuda-gdb

- This session contains two codes which you will modify
  - vec_add-debug-printf.cu
  - vec_add-debug-memcheck.cu
- And one you should not modify
  - check-debug-printf.hpp
- To compile just use make
- Run first code
  - srun ./vec_add-debug-printf.exe

```
[...]
idx(...) total_threads(...) strides_per_threads(...)
before i(31) A(31) B(31) C(0)
after i(31) A(31) B(31) C(62)
error in: 32 / 64 cpu 0 gpu
error in: 33 / 66 cpu 0 gpu
error in: 34 / 68 cpu 0 gpu
        Correctness Test Failed
```



```
[...]
idx(...) total_threads(...) strides_per_threads(...)
before i(31) A(31) B(31) C(0)
after i(31) A(31) B(31) C(62)
        Correctness Test Passed
```

Cuda-gdb

- Run first code
  - srun cuda-memcheck ./vec_add-debug-memcheck.exe

```
========= CUDA-MEMCHECK
========= Invalid __global__ read of size 4
=========     at 0x00000e20 in /global/u1/h/hbrunie/nersc_cuda_tutorial/Session-3/vec_add-debug-memcheck.cu:66:vector_
=========     by thread (7,0,0) in block (3,0,0)
=========     Address 0x2aaae24000fc is out of bounds
=========     Device Frame:/global/u1/h/hbrunie/nersc_cuda_tutorial/Session-3/vec_add-debug-memcheck.cu:66:vector_add_
=========     Saved host backtrace up to driver entry point at kernel launch time
=========     Host Frame:/usr/lib64/libcuda.so.1 (cuLaunchKernel + 0x346) [0x297db6]
=========     Host Frame:./vec_add-debug-memcheck.exe [0x165e9]
=========     Host Frame:./vec_add-debug-memcheck.exe [0x16677]
=========     Host Frame:./vec_add-debug-memcheck.exe [0x4c9d5]
=========     Host Frame:./vec_add-debug-memcheck.exe [0x4713]
=========     Host Frame:./vec_add-debug-memcheck.exe (_Z52__device_stub__Z24vector_add_kernel_memoryPKiS0_PiiiPKiS0_P
=========     Host Frame:./vec_add-debug-memcheck.exe (_Z24vector_add_kernel_memoryPKiS0_Piii + 0x38) [0x45f7]
=========     Host Frame:./vec_add-debug-memcheck.exe (main + 0x263) [0x42f5]
=========     Host Frame:/lib64/libc.so.6 (__libc_start_main + 0xea) [0x20f8a]
=========     Host Frame:./vec_add-debug-memcheck.exe (_start + 0x2a) [0x3e9a]
=========

[...]
```

- Inside gdb promt
  - cuda kernel 0 thread (7,0,0) block (0,0,0)

Cuda-gdb

- Run first code
  - srun --pty cuda-gdb ./vec_add-debug-memcheck.exe

```
(cuda-gdb) b vector_add_kernel_memory
Breakpoint 2 at 0x4045da: file vec_add-debug-memcheck.cu, line 50.
(cuda-gdb) c
Continuing.
[New Thread 0x2aaab3378700 (LWP 55823)]
[New Thread 0x2aaab3579700 (LWP 55824)]
[Switching focus to CUDA kernel 0, grid 1, block (0,0,0), thread (0,0,0), dev

Thread 1 "vec_add-debug-m" hit Breakpoint 2, vector_add_kernel_memory<<<(4,1,
    C=0x2aaae1c00400, size=35, stride_size=1) at vec_add-debug-memcheck.cu:51
51        int idx = threadIdx.x + blockIdx.x * blockDim.x;
(cuda-gdb) cuda kernel 0 block (0,0,0) thread (3,0,0)
[Switching focus to CUDA kernel 0, grid 1, block (0,0,0), thread (3,0,0), dev
51        int idx = threadIdx.x + blockIdx.x * blockDim.x;
(cuda-gdb) n
52        int total_threads = gridDim.x*blockDim.x;
(cuda-gdb) p idx
$1 = 3
(cuda-gdb) p total_threads
$2 = <optimized out>
(cuda-gdb) p gridDim.x*blockDim.x
$3 = 32
(cuda-gdb) n
54        int strides_per_thread = size/(total_threads*stride_size);
(cuda-gdb) n
55        if(idx< (total_threads*stride_size % size))
(cuda-gdb) p total_threads*stride_size
$4 = 32
```

Thank you!

I thank Woo-Sun Yang for his slide on Debugging on GPU (02-28-2020)