Parallel Programming

Lab4 2018/12/6

Overall Performance Optimization Strategies

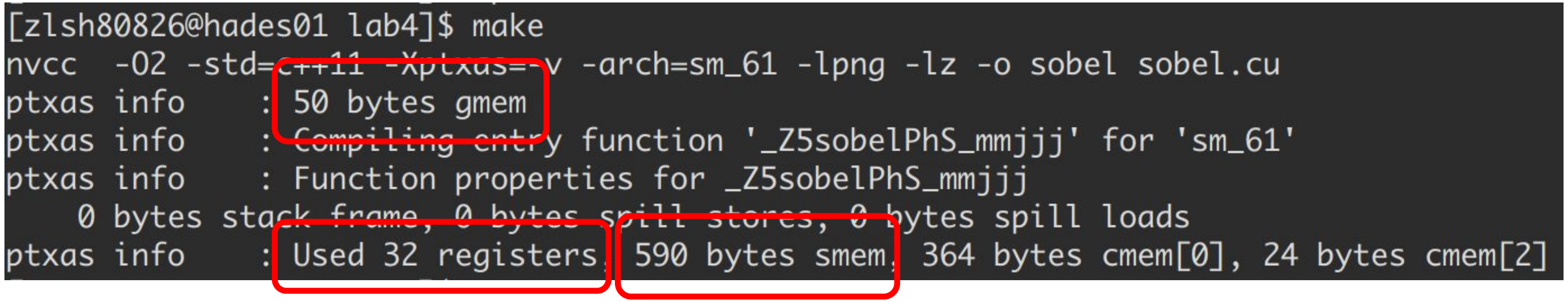
- ❖ Maximize parallel execution to achieve maximum utilization
- ❖ Optimize memory usage to achieve maximum memory throughput
- ❖ Optimize instruction usage to achieve maximum instruction throughput

CUDA Optimization technique

- ❖ Resource infomation & device information
- ❖ 2D data placement
- ❖ Share memory
- ❖ Streaming

Resource information & device information

- ❖ Device information -> deviceQuery
- ❖ Resource information
  - ➢ add “-Xptxas=-v” to compiler flag to check the resource allocation
  - ➢ add “-arch=sm_61” to make the compiler allocate suitable resource (default is sm_30)



```
[zlsh80826@hades01 lab4]$ make
nvcc  -O2 -std=c++11 -Xptxas=-v -arch=sm_61 -lpng -lz -o sobel sobel.cu
ptxas info    : 50 bytes gmem
ptxas info    : Compiling entry function '_Z5sobelPhS_mmjjj' for 'sm_61'
ptxas info    : Function properties for _Z5sobelPhS_mmjjj
    0 bytes stack frame, 0 bytes spill stores, 0 bytes spill loads
ptxas info    : Used 32 registers, 590 bytes smem, 364 bytes cmem[0], 24 bytes cmem[2]
```

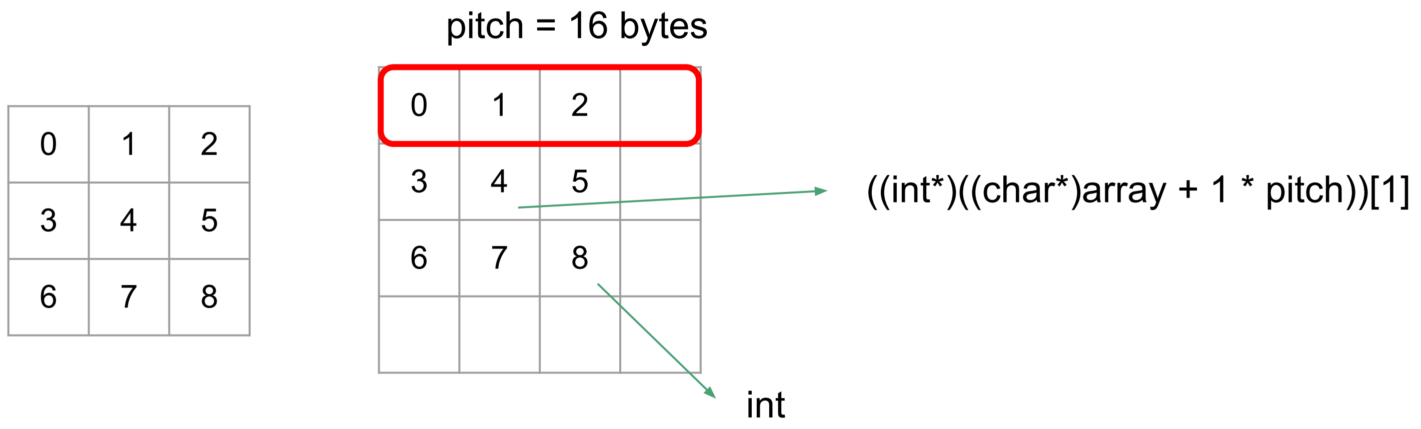
Occupancy

- With the above information, we can calculate how many blocks can concurrently run on a SM
- Number of active blocks are limited by
  - Shared memory
  - Register
  - Max threads/threads per block
- We can calculate the occupancy manually or API
  - __host__ __device__ cudaError_t cudaOccupancyMaxActiveBlocksPerMultiprocessor (int* numBlocks, const void* func, int blockSize, size_t dynamicSMemSize)
  - Reference

2D Data placement

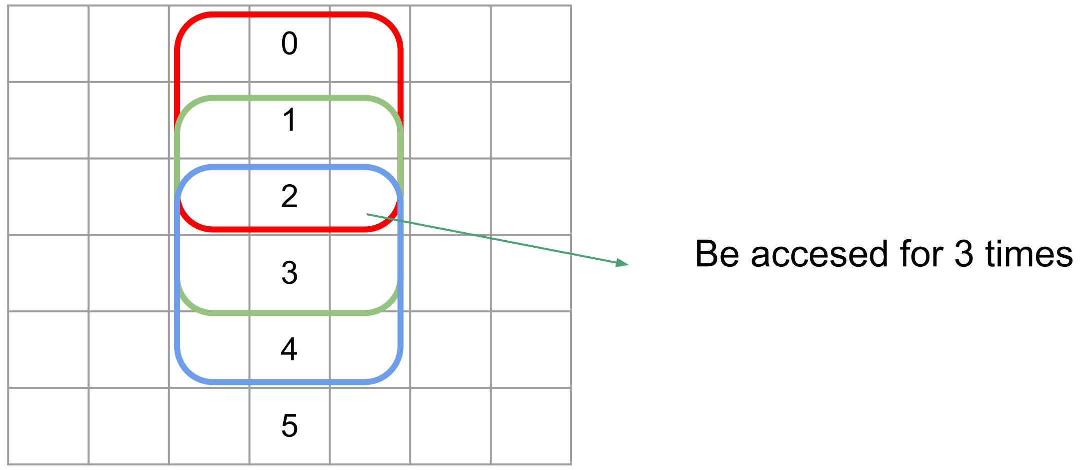
- Most of time we use GPU devices to solve the 2D, 3D problems
- Official document suggest using 2D API for 2D memory operation
- Use cudaMallocPitch to allocate memory
- cudaMemcpy2D for memory copy

```
T* pElement = (T*)((char*)BaseAddress + Row * pitch) + Column
```



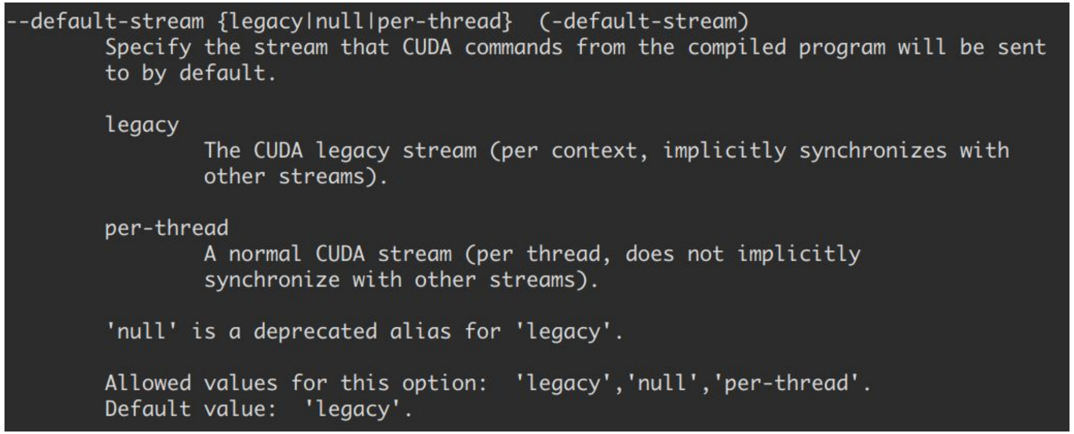
Share memory

- Share memory is faster than device memory
- Share memory take advantages on the appications with data locality
- Example for sobel.

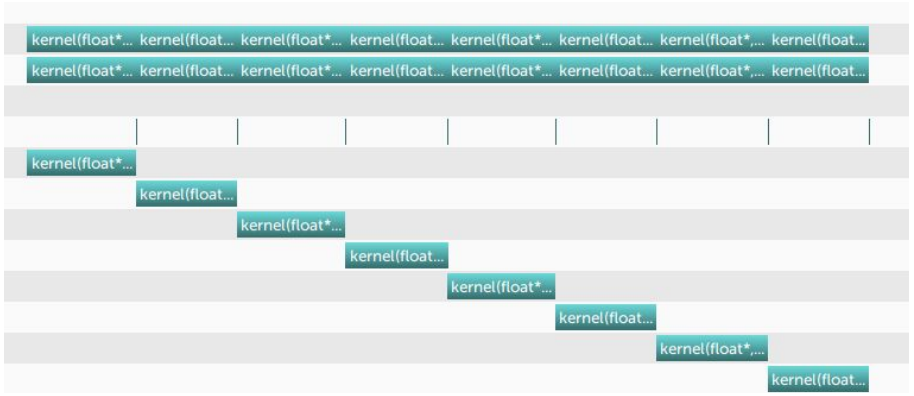


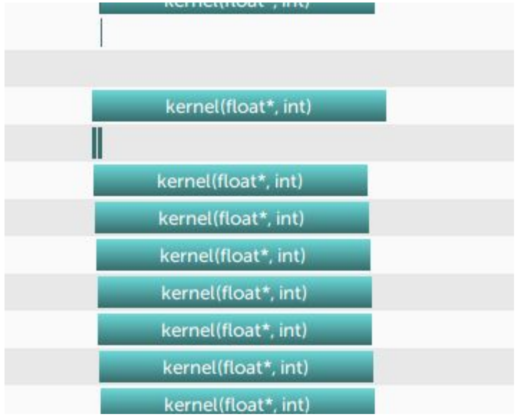
Streaming

❖ Reference



```
--default-stream {legacy|null|per-thread}  (-default-stream)
        Specify the stream that CUDA commands from the compiled program will be sent
        to by default.

        legacy
                The CUDA legacy stream (per context, implicitly synchronizes with
                other streams).

        per-thread
                A normal CUDA stream (per thread, does not implicitly
                synchronize with other streams).

        'null' is a deprecated alias for 'legacy'.

        Allowed values for this option:  'legacy','null','per-thread'.
        Default value:  'legacy'.
```





Hints

- ❖ Easy
  - ➢ Calculate the occupancy and tune the [thread per block]
  - ➢ Compare the speed
    - ■ mask on share memory, const memory, device memory
  - ➢ Images are 2D (2D API)
  - ➢ Take out branch (padding with 0)
- ❖ Hard
  - ➢ Streaming
  - ➢ Slides page 7.

0	1	2	3
4	5	6	7
8	9	10	11

		0	1	2	3		
		4	5	6	7		
		8	9	10	11		

Lab4 Assignment

- Optimize sobel.cu with any optimization techniques you like
- We still provide modified cpu code
  - cp /home/pp18/shared/lab4 ~/homework
  - make
  - srun -n 1 --gres=gpu:1 -ppp ./sobel <input> <output>
  - lab4-judge for judging
- Submission
  - After 12/6 23:59, We only use the scoreboard for scoring
- Deadline
  - 12/9 23:59

Grading

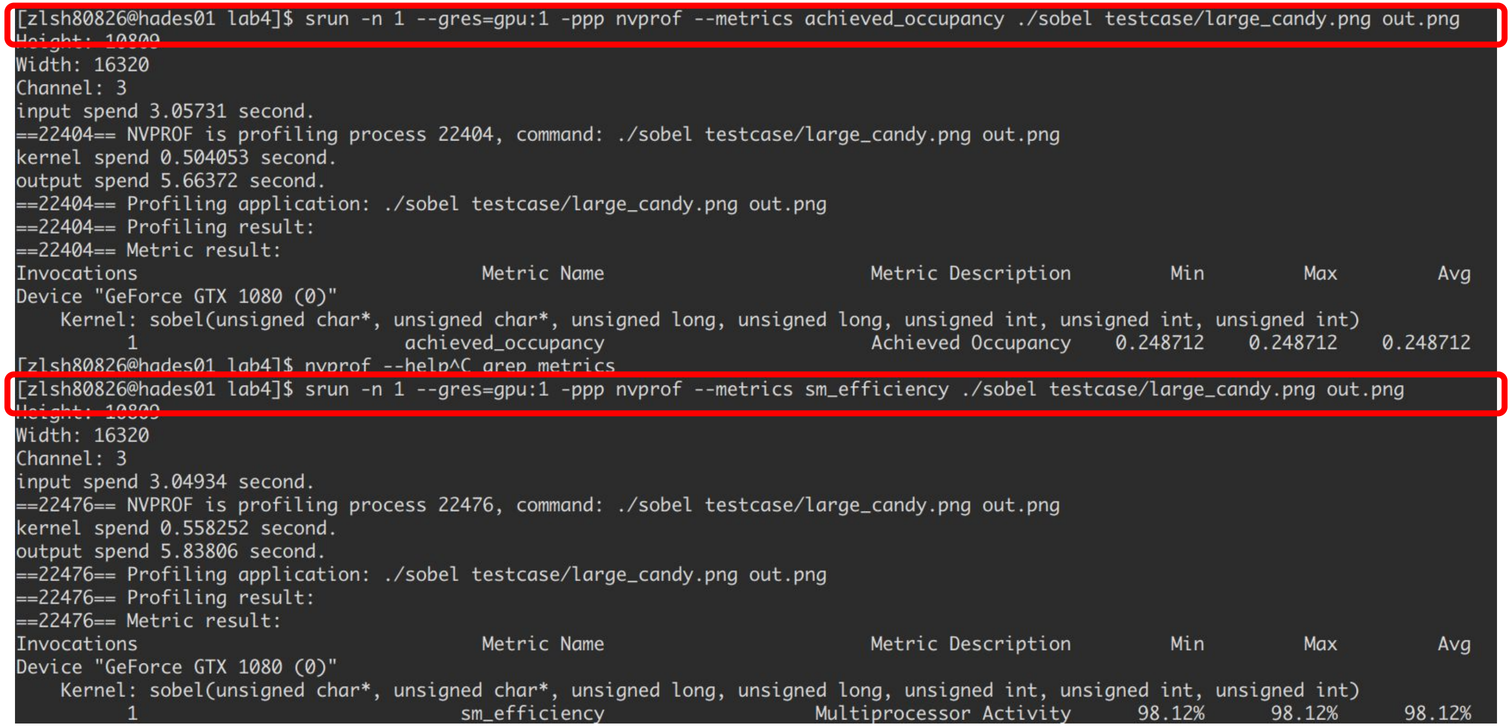
	Before 12/6 23:59	Before deadline
Faster than TA	150	125
Faster than benchmark100	100	100
Faster than benchmark60	60	60

Port the sobel on GPU and tune the threadPerBlock

- In lab(12/6 23:59) only
  - If you can’t faster than mewtwo, demo the easy part of the hints can also get 100

Appendix: nvprof

metrics



```
[zlsh80826@hades01 lab4]$ srun -n 1 --gres=gpu:1 -ppp nvprof --metrics achieved_occupancy ./sobel testcase/large_candy.png out.png
Height: 10809
Width: 16320
Channel: 3
input spend 3.05731 second.
==22404== NVPROF is profiling process 22404, command: ./sobel testcase/large_candy.png out.png
kernel spend 0.504053 second.
output spend 5.66372 second.
==22404== Profiling application: ./sobel testcase/large_candy.png out.png
==22404== Profiling result:
==22404== Metric result:
Invocations                               Metric Name                        Metric Description         Min         Max         Avg
Device "GeForce GTX 1080 (0)"
    Kernel: sobel(unsigned char*, unsigned char*, unsigned long, unsigned long, unsigned int, unsigned int, unsigned int)
          1                          achieved_occupancy                        Achieved Occupancy    0.248712    0.248712    0.248712
[zlsh80826@hades01 lab4]$ nvprof --help^C grep metrics
[zlsh80826@hades01 lab4]$ srun -n 1 --gres=gpu:1 -ppp nvprof --metrics sm_efficiency ./sobel testcase/large_candy.png out.png
Height: 10809
Width: 16320
Channel: 3
input spend 3.04934 second.
==22476== NVPROF is profiling process 22476, command: ./sobel testcase/large_candy.png out.png
kernel spend 0.558252 second.
output spend 5.83806 second.
==22476== Profiling application: ./sobel testcase/large_candy.png out.png
==22476== Profiling result:
==22476== Metric result:
Invocations                               Metric Name                        Metric Description         Min         Max         Avg
Device "GeForce GTX 1080 (0)"
    Kernel: sobel(unsigned char*, unsigned char*, unsigned long, unsigned long, unsigned int, unsigned int, unsigned int)
          1                               sm_efficiency                   Multiprocessor Activity      98.12%      98.12%      98.12%
```