Background and introduction

使用 *Assess, Parallelize, Optimize, Deploy (APOD)* design cycle 可以快速找到应用中可以从gpu获得最大收益的部分。本文以华星光电项目为例，记录了一个简单的加速过程。https://docs.nvidia.com/cuda/cuda-c-best-practices-guide/index.html

assess: 优化前sdk性能

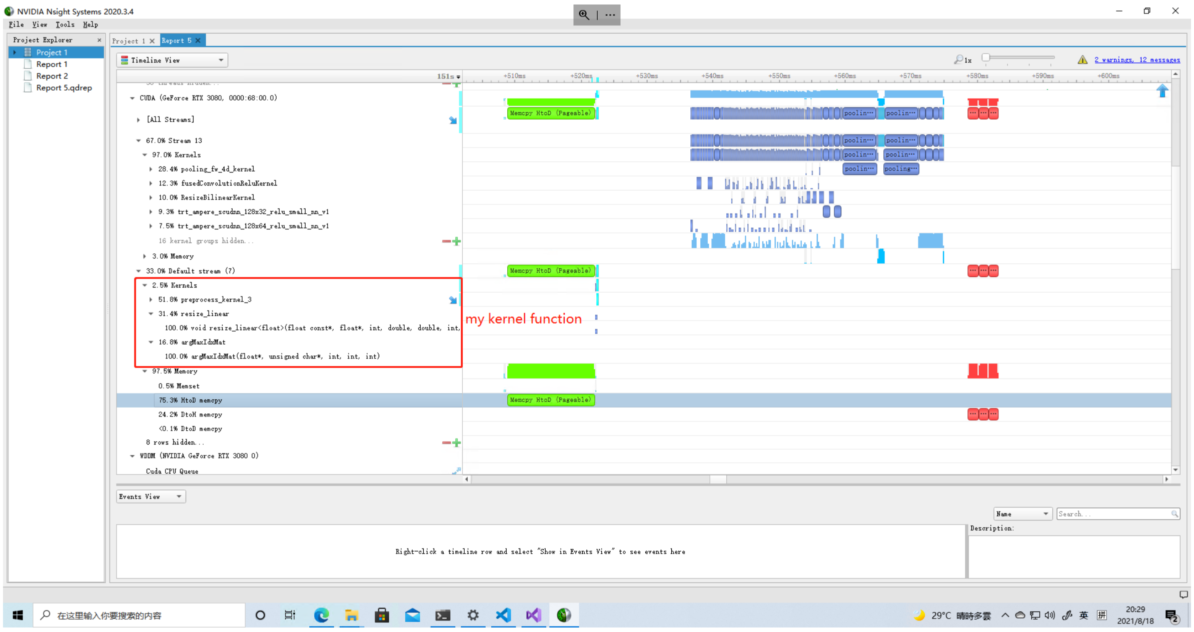
这一步评估程序的加速潜力，使用weak_scaling 来估计出理论上最高的加速比。当每个处理器的工作量基本固定，评估计算时间如何随着处理器数量变化。

三个hotspot:

- 模型初始化加速 （40% 时间）
- 将前处理的resize函数使用gpu并行
- 去除多余的 cudaDeviceSynchronize 调用

parallelize and optimize (cuda并行方案)

当完成后两项后，加速效果并不理想，如下图所示：



```
▾ CUDA (GeForce RTX 3080, 0000:68:00.0)
  ▸ [All Streams]
  ▾ 67.0% Stream 13
    ▾ 97.0% Kernels
      ▸ 28.4% pooling_fw_4d_kernel
      ▸ 12.3% fusedConvolutionReluKernel
      ▸ 10.0% ResizeBilinearKernel
      ▸ 9.3% trt_ampere_scudnn_128x32_relu_small_nn_v1
      ▸ 7.5% trt_ampere_scudnn_128x64_relu_small_nn_v1
        16 kernel groups hidden...
    ▸ 3.0% Memory
  ▾ 33.0% Default stream (7)
    ▾ 2.5% Kernels
      ▸ 51.8% preprocess_kernel_3
      ▾ 31.4% resize_linear
          100.0% void resize_linear<float>(float const*, float*, int, double, double, int,
      ▾ 16.8% argMaxIdxMat
          100.0% argMaxIdxMat(float*, unsigned char*, int, int, int)
    ▾ 97.5% Memory
        0.5% Memset
        75.3% HtoD memcpy
        24.2% DtoH memcpy
        <0.1% DtoD memcpy
  8 rows hidden...
▾ WDDM (NVIDIA GeForce RTX 3080 0)
  Cuda CPU Queue
```

Right-click a timeline row and select "Show in Events View" to see events here

此处可以发现几个问题

1. 模型初始化和推理的时间较长
2. 大量的时间（14ms）花费在了拷贝内容到显存上和从显存拷贝内容。

Deploy: 优化后sdk性能

备注:

Strong scaling and Amdahl's Law

Strong scaling is a measure of how, for a fixed overall problem size, the time to solutiondecreases as more processors are added to a system. An application that exhibits linearstrong scaling has a speedup equal to the number of processors used.

$$S = \frac{1}{(1+P) + \frac{P}{N}}$$

Here P is the fraction of the total serial execution time taken by the portion of code that can be parallelized and N is the number of processors over which the parallel portion of
the code runs.

Weak Scaling and Gustafson's Law

Weak scaling is a measure of how the time to solution changes as more processors are added to a system with a fixed problem size per processor.

$$S = N + (1-P)(1-N) = (1-P) + PN$$

Here P is the fraction of the total serial execution time taken by the portion of code that can be parallelized and N is the number of processors over which the parallel portion of the code runs