Лабораторная 7

(методические указания)

Задание.
Реализуйте транспонирование матрицы размерностью N*K без использования разделяемой памяти, с разделяемой памятью без разрешения конфликта банков и с разрешением конфликта банков. Сравните время выполнения соответствующих ядер на GPU. Для всех трёх случаев определите эффективность использования разделяемой памяти с помощью метрик nvprof или ncu.

Цель: приобретение навыков использования разделяемой памяти.

I. Подготовить ядра для тестирования

```
_global__ void gInit(float* a){
  int k=threadIdx.x+blockIdx.x*blockDim.x;
  int n=threadIdx.y+blockIdx.y*blockDim.y;
  int K=blockDim.x*gridDim.x;

  a[k+n*K]=(float)(k+n*K);
}
__global__ void gTranspose1(float* a, float* b){
  int k=threadIdx.x+blockIdx.x*blockDim.x;
  int n=threadIdx.y+blockIdx.y*blockDim.y;
  int K=blockDim.x*gridDim.x;
  int N=blockDim.y*gridDim.y;

  b[k+n*K]=a[n+k*N];
}
```

```
__global__ void gTranspose2(float* a, float* b){
  int k=threadIdx.x+blockIdx.x*blockDim.x;
  int n=threadIdx.y+blockIdx.y*blockDim.y;
  int K=blockDim.x*gridDim.x;
  int N=blockDim.y*gridDim.y;

  b[n+k*N]=a[k+n*K];
}
```

```
#define SH_DIM 32
__global__ void gTransposeSM(float* a, float* b){
  __shared__ float cache[SH_DIM][SH_DIM];
  int k=threadIdx.x+blockIdx.x*blockDim.x;
  int n=threadIdx.y+blockIdx.y*blockDim.y;
  int N=blockDim.x*gridDim.x;

  cache[threadIdx.y][threadIdx.x]=a[k+n*N];
  __syncthreads();

  k=threadIdx.x+blockIdx.y*blockDim.x;
  n=threadIdx.y+blockIdx.x*blockDim.y;
  b[k+n*N]=cache[threadIdx.x][threadIdx.y];
}
```

```
__global__ void gTransposeSM_WC(float* a, float* b){
  __shared__ float cache[SH_DIM][SH_DIM+1];
  int k=threadIdx.x+blockIdx.x*blockDim.x;
  int n=threadIdx.y+blockIdx.y*blockDim.y;
  int N=blockDim.x*gridDim.x;

  cache[threadIdx.y][threadIdx.x]=a[k+n*N];
  __syncthreads();

  k=threadIdx.x+blockIdx.y*blockDim.x;
  n=threadIdx.y+blockIdx.x*blockDim.y;
  b[k+n*N]=cache[threadIdx.x][threadIdx.y];
}
```

II. Написать драйвер для тестирования

```
#include <stdio.h>
#define CUDA_CHECK_RETURN(value) {\
       cudaError_t    _m_cudaStat = value;\
       if (_m_cudaStat != cudaSuccess) {\
         fprintf(stderr, "Error %s at line %d in file %s\n",\
         cudaGetErrorString(_m_cudaStat), __LINE__, __FILE__);\
         exit(1);\
       } }
```

```
int main(int argc, char* argv[]){
    if(argc<3){
       fprintf(stderr, "USAGE: tr_mat-25  <dimension of matrix> <dimension of
                                                                    threads>\n");
       return -1;
    }
    int N=atoi(argv[1]);
    int dim_of_threads=atoi(argv[2]);
    if(N%dim_of_threads){
      fprintf(stderr, "change dimensions\n");
      return -1;
    }
    int dim_of_blocks=N/dim_of_threads;
    const int max_size=1<<8;
    if(dim_of_blocks>max_size){
      fprintf(stderr, "too many blocks\n");
      return -1;
    }
```

```
float *a, *b;

 cudaMalloc((void**)&a, N*N*sizeof(float));
 cudaMalloc((void**)&b, N*N*sizeof(float));

 gInit<<<dim3(dim_of_blocks, dim_of_blocks),
         dim3(dim_of_threads,dim_of_threads)>>>(a);
 cudaDeviceSynchronize();
 CUDA_CHECK_RETURN(cudaGetLastError());

 cudaMemset(b, 0, N*N*sizeof(float));
```

```
    gTranspose1<<<dim3(dim_of_blocks, dim_of_blocks),
                dim3(dim_of_threads,dim_of_threads)>>>(a,b);
    cudaDeviceSynchronize();  CUDA_CHECK_RETURN(cudaGetLastError());
    gTranspose2<<<dim3(dim_of_blocks, dim_of_blocks),
                dim3(dim_of_threads,dim_of_threads)>>>(a,b);
    cudaDeviceSynchronize();  CUDA_CHECK_RETURN(cudaGetLastError());
    gTransposeSM<<<dim3(dim_of_blocks, dim_of_blocks),
                dim3(dim_of_threads,dim_of_threads)>>>(a,b);
    cudaDeviceSynchronize();  CUDA_CHECK_RETURN(cudaGetLastError());
    gTransposeSM_WC<<<dim3(dim_of_blocks, dim_of_blocks),
                dim3(dim_of_threads,dim_of_threads)>>>(a,b);
    cudaDeviceSynchronize(); CUDA_CHECK_RETURN(cudaGetLastError());

    cudaFree(a);
    cudaFree(b);
}
```

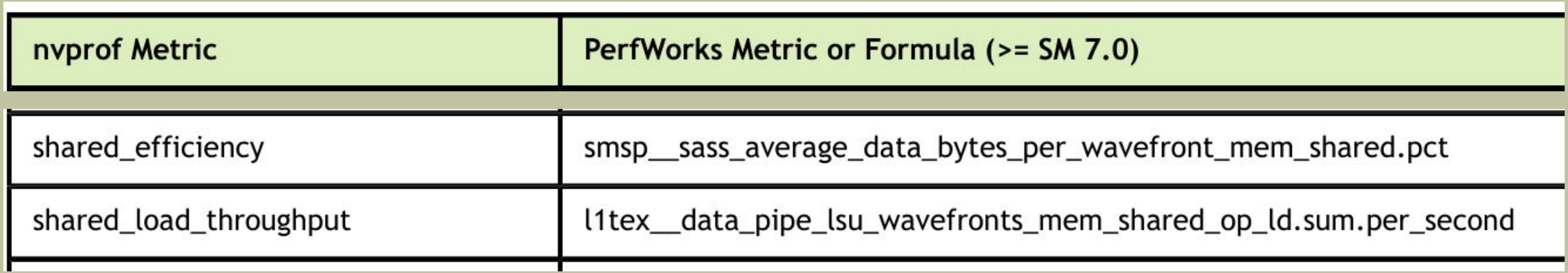
III. Подобрать метрики для анализа производительности

NSIGHT COMPUTE COMMAND LINE INTERFACE

v2023.3.1 | October 2023

User Manual

6.3. Metric Comparison



nvprof Metric	PerfWorks Metric or Formula (>= SM 7.0)
shared_efficiency	smsp__sass_average_data_bytes_per_wavefront_mem_shared.pct
shared_load_throughput	l1tex__data_pipe_lsu_wavefronts_mem_shared_op_ld.sum.per_second

III. Провести профилирование

```
Lab6> nvprof  ./tr_mat25 256 32
==9499== NVPROF is profiling process 9499, command: ./tr_mat25 256 32
==9499== Profiling application: ./tr_mat25 256 32
==9499== Profiling result:
           Type  Time(%)      Time     Calls       Avg       Min       Max  Name
GPU activities:   29.34%  9.9530us         1  9.9530us  9.9530us  9.9530us
gTranspose2(float*, float*)
                  22.36%  7.5840us         1  7.5840us  7.5840us  7.5840us
gTransposeSM(float*, float*)
                  21.22%  7.1990us          1  7.1990us  7.1990us  7.1990us
gTranspose1(float*, float*)
                  11.04%  3.7440us          1  3.7440us  3.7440us  3.7440us
gInit(float*)
                   9.91%  3.3600us           1  3.3600us  3.3600us  3.3600us
gTransposeSM_WC(float*, float*)
```

```
...> nvprof -m shared_efficiency ./tr_mat25 256 32
Invocations            Metric Name                  Metric Description
                 Min         Max                Avg
Device "GeForce GTX 560 Ti (0)"
Kernel: gTranspose1(float*, float*)
      1                shared_efficiency         Shared Memory Efficiency
                 0.00%       0.00%              0.00%
Kernel: gTranspose2(float*, float*)
                  0.00%       0.00%              0.00%
Kernel: gTransposeSM(float*, float*)
                 6.06%        6.06%              6.06%
Kernel: gTransposeSM_WC(float*, float*)
               100.00%    100.00%          100.00%
```

```
/Lab6> ncu --metrics
smsp__sass_average_data_bytes_per_wavefront_mem_shared.pct ./tr_mat25 256 32
```

```
gTranspose1(float *, float *) (8, 8, 1)x(32, 32, 1), Context 1, Stream 7, Device 0, CC 7.5
  Section: Command line profiler metrics
  ---------------------------------------------------------- ----------- ------------
  Metric Name                                                Metric Unit Metric Value
  ---------------------------------------------------------- ----------- ------------
  smsp__sass_average_data_bytes_per_wavefront_mem_shared.pct           %            0
  ---------------------------------------------------------- ----------- ------------

gTranspose2(float *, float *) (8, 8, 1)x(32, 32, 1), Context 1, Stream 7, Device 0, CC 7.5
  Section: Command line profiler metrics
  ---------------------------------------------------------- ----------- ------------
  Metric Name                                                Metric Unit Metric Value
  ---------------------------------------------------------- ----------- ------------
  smsp__sass_average_data_bytes_per_wavefront_mem_shared.pct           %            0
  ---------------------------------------------------------- ----------- ------------
```

```
gTransposeSM(float *, float *) (8, 8, 1)x(32, 32, 1), Context 1, Stream 7, Device 0, CC 7.5
   Section: Command line profiler metrics
   ----------------------------------------------------- ---------- -----------
   Metric Name                                           Metric Unit Metric Value
   ----------------------------------------------------- ---------- -----------
   smsp__sass_average_data_bytes_per_wavefront_mem_shared.pct          %        6,06
   ----------------------------------------------------- ---------- -----------

 gTransposeSM_WC(float *, float *) (8, 8, 1)x(32, 32, 1), Context 1, Stream 7, Device 0, CC 7.5
   Section: Command line profiler metrics
   ----------------------------------------------------- ---------- -----------
   Metric Name                                           Metric Unit Metric Value
   ----------------------------------------------------- ---------- -----------
   smsp__sass_average_data_bytes_per_wavefront_mem_shared.pct          %         100
   ----------------------------------------------------- ---------- -----------
```