

Лабораторная 2

Предварительная работа: преподаватель знакомит студентов с обработкой ошибок с помощью CUDA API и профилированием кода с помощью Events - объектов CUDA разбирая программы ПРИЛОЖЕНИЯ 1. Преподаватель знакомит студентов с профилировщиком NVIDIA — nvprof (ПРИЛОЖЕНИЕ 2).

Задание: выполнить задание лабораторной 1, используя Events и nvprof.

Цель: научиться обрабатывать ошибки и профилировать код при выполнении программы на GPU.

ПРИЛОЖЕНИЕ 1

//TestErr.cu

```
#include <cuda.h>
```

```
#include <stdio.h>
```

```
#define CUDA_CHECK_RETURN(value) {\n    cudaError_t _m_cudaStat = value;\n    if (_m_cudaStat != cudaSuccess) {\n        fprintf(stderr, "Error %s at line %d in file %s\\n",\n            cudaGetErrorString(_m_cudaStat), __LINE__, __FILE__);\n        exit(1);\n    } } //макрос для обработки ошибок
```

```
__global__ void gTest(float* a){\n    a[threadIdx.x+blockDim.x*blockIdx.x]=(float)(threadIdx.x+blockDim.x*blockIdx.x);\n}\nint main(){
```

```
    float *da, *ha;
```

```
    int num_of_blocks=10, threads_per_block= 1025;
```

```
    int N=num_of_blocks*threads_per_block;
```

```
    ha=(float*)calloc(N, sizeof(float));
```

```
    CUDA_CHECK_RETURN(cudaMalloc((void**)&da,N*sizeof(float)));
```

```
    gTest<<<dim3(num_of_blocks), dim3(threads_per_block)>>>(da);
```

```
    // cudaThreadSynchronize();
```

```
    CUDA_CHECK_RETURN(cudaDeviceSynchronize());
```

```
    CUDA_CHECK_RETURN(cudaGetLastError());
```

```
    CUDA_CHECK_RETURN(cudaMemcpy(ha,da,N*sizeof(float), cudaMemcpyDeviceToHost));
```

```
    for(int i=0;i<N;i++)\n        printf("%g\\n",ha[i]);
```

```
    free(ha);
```

```
    cudaFree(da);
```

```
    return 0;
```

```
}
```

```

//TestEvent.cu
#include <cuda.h>
#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);\
    } }

__global__ void gTest(float* a){
    a[threadIdx.x+blockDim.x*blockIdx.x]=(float)(threadIdx.x+blockDim.x*blockIdx.x);
}

int main(){

    float *da, *ha;
    int num_of_blocks=10000, threads_per_block=1024;
    int N=num_of_blocks*threads_per_block;

    float elapsedTime;
    cudaEvent_t start,stop; // встроенный тип данных – структура, для фиксации контрольных
                           //точек
    cudaEventCreate(&start); // инициализация
    cudaEventCreate(&stop); // событий

    ha=(float*)calloc(N, sizeof(float));
    CUDA_CHECK_RETURN(cudaMalloc((void**)&da,N*sizeof(float)));

    cudaEventRecord(start,0); // привязка события
    gTest<<<dim3(num_of_blocks), dim3(threads_per_block)>>>(da);
    cudaEventRecord(stop,0); // привязка события
    cudaEventSynchronize(stop); // синхронизация по событию
    // cudaThreadSynchronize();
    //CUDA_CHECK_RETURN(cudaDeviceSynchronize());
    CUDA_CHECK_RETURN(cudaGetLastError());
    cudaEventElapsedTime(&elapsedTime,start,stop); // вычисление затраченного времени

    fprintf(stderr,"gTest took %g\n", elapsedTime);

    cudaEventDestroy(start); // освобождение
    cudaEventDestroy(stop); // памяти

    CUDA_CHECK_RETURN(cudaMemcpy(ha,da,N*sizeof(float), cudaMemcpyDeviceToHost));
    for(int i=0;i<N;i++)
        printf("%g\n",ha[i]);

    free(ha);

```

```
cudaFree(da);
```

```
return 0;  
}
```

ПРИЛОЖЕНИЕ 2

Кол-во нитей - 10*1024

```
> nvprof ./Lab2-1 > tmp  
==8454== NVPROF is profiling process 8454, command: ./Lab2-1  
gTest took 0.034144  
==8454== Profiling application: ./Lab2-1  
==8454== Profiling result:  
Time(%)   Time    Calls   Avg     Min    Max  Name  
70.99%  7.4240us      1  7.4240us  7.4240us  7.4240us [CUDA memcpy DtoH]  
29.01%  3.0340us      1  3.0340us  3.0340us  3.0340us gTest(float*)  
  
==8454== API calls:  
Time(%)   Time    Calls   Avg     Min    Max  Name  
99.03%  64.094ms       1  64.094ms  64.094ms  64.094ms cudaMalloc  
0.57%   369.63us     83  4.4530us   235ns  158.67us cuDeviceGetAttribute  
0.12%   79.077us      1  79.077us  79.077us  79.077us cudaFree  
0.08%   50.090us      1  50.090us  50.090us  50.090us cuDeviceTotalMem  
0.06%   39.709us      1  39.709us  39.709us  39.709us cuDeviceGetName  
0.05%   34.740us      1  34.740us  34.740us  34.740us cudaMemcpy  
0.03%   22.301us      1  22.301us  22.301us  22.301us cudaLaunch  
0.01%    8.9780us     2  4.4890us  3.1640us  5.8140us cudaEventRecord  
0.01%    4.4930us     2  2.2460us   914ns  3.5790us cudaEventCreate  
0.01%    3.6370us      1  3.6370us  3.6370us  3.6370us cudaEventSynchronize  
0.01%    3.5370us     2  1.7680us   917ns  2.6200us cudaEventDestroy  
0.00%    3.2010us      1  3.2010us  3.2010us  3.2010us cudaSetupArgument  
0.00%    1.8790us      2    939ns   426ns  1.4530us cuDeviceGetCount  
0.00%    1.8130us      1  1.8130us  1.8130us  1.8130us cudaEventElapsedTime  
0.00%    1.6770us      1  1.6770us  1.6770us  1.6770us cudaConfigureCall  
0.00%    1.1350us      2    567ns   300ns   835ns cuDeviceGet  
0.00%     511ns      1    511ns   511ns   511ns cudaGetLastError
```

Кол-во нитей - 100*1024

```
> nvprof ./Lab2-1 > tmp  
==11823== NVPROF is profiling process 11823, command: ./Lab2-1  
gTest took 0.039168  
==11823== Profiling application: ./Lab2-1  
==11823== Profiling result:  
Time(%)   Time    Calls   Avg     Min    Max  Name  
80.30%  62.589us      1  62.589us  62.589us  62.589us [CUDA memcpy DtoH]  
19.70%  15.359us      1  15.359us  15.359us  15.359us gTest(float*)
```

Кол-во нитей - 1000*1024

```
> nvprof ./Lab2-1 > tmp
==12723== NVPROF is profiling process 12723, command: ./Lab2-1
gTest took 0.168256
==12723== Profiling application: ./Lab2-1
==12723== Profiling result:
Time(%)   Time    Calls    Avg     Min     Max  Name
90.72%  1.3892ms      1  1.3892ms  1.3892ms  1.3892ms  [CUDA memcpy DtoH]
 9.28%  142.13us      1  142.13us  142.13us  142.13us  gTest(float*)
```

Кол-во нитей - 10000*1024

```
> nvprof ./Lab2-1 > tmp
==12579== NVPROF is profiling process 12579, command: ./Lab2-1
gTest took 1.44032
==12579== Profiling application: ./Lab2-1
==12579== Profiling result:
Time(%)   Time    Calls    Avg     Min     Max  Name
88.93%  11.366ms      1  11.366ms  11.366ms  11.366ms  [CUDA memcpy DtoH]
11.07%  1.4144ms      1  1.4144ms  1.4144ms  1.4144ms  gTest(float*)
```