

# Parallel Homework #3

刘康来

2021 年 5 月 18 日

图 1: Here is the hardware's information

```
CPU: Intel i7-8565U (8) @ 4.600GHz
GPU: Intel UHD Graphics 620
GPU: NVIDIA GeForce MX250
Memory: 4971MiB / 7708MiB
```

图 2: About the GPU and CUDA

```
→ ~ nvidia-smi
Mon May 17 17:59:32 2021

+-----+
| NVIDIA-SMI 465.27      Driver Version: 465.27      CUDA Version: 11.3      |
+-----+-----+
| GPU Name      Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC | |
| Fan  Temp  Perf  Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|               |                  |              | MIG M. |
+-----+-----+
| 0  NVIDIA GeForce ...  Off  | 00000000:02:00:0 | Off      | N/A |
| N/A   39C    P0     N/A /  N/A | 0MiB / 2002MiB |      1%    Default  |
|               |                  |              |      |
+-----+-----+

+-----+
| Processes: |
| GPU  GI    CI          PID    Type    Process name                  GPU Memory |
|      ID    ID                                   |           Usage          |
+-----+-----+
| No running processes found |
+-----+
```

图 3: Here is the test runnig

```
→ makeVer git:(main) × ./gemm_test 1000 1000 1000
Matrix A is 1000 x 1000, matrix B is 1000 x 1000

GEMM (gemm_openMP)(row-col, A and B are in row-major)) used 0.24020 s, 8.33 GFlop/s
GEMM (gemm_openMP)(row-col, A and B are in row-major) PASS!

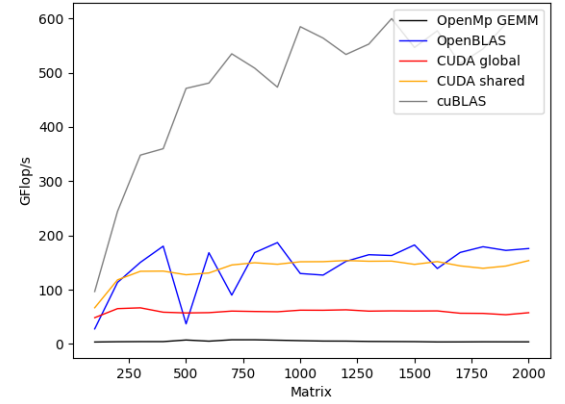
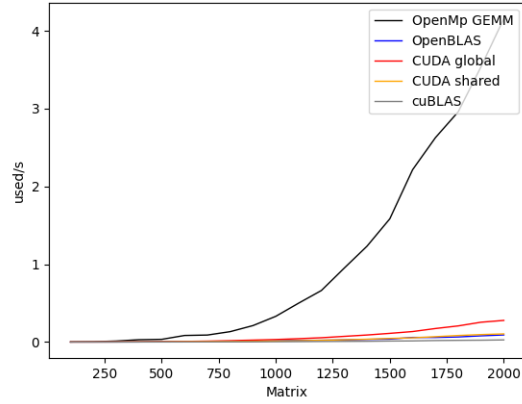
GEMM (OpenBLAS)(row-col, A and B are in row-major)) used 0.00818 s, 244.38 GFlop/s
GEMM (OpenBLAS)(row-col, A and B are in row-major) PASS!

GEMM (cuda_global)(row-col, A and B are in row-major)) used 0.03186 s for 10 bench(s) in average, 62.78 GFlop/s
GEMM (cuda_global)(row-col, A and B are in row-major) PASS!

GEMM (cuda_shared)(row-col, A and B are in row-major)) used 0.01314 s for 10 bench(s) in average, 152.24 GFlop/s
GEMM (cuda_shared)(row-col, A and B are in row-major) PASS!

GEMM (cublas)(row-col, A and B are in row-major)) used 0.00341 s for 10 bench(s) in average, 586.40 GFlop/s
GEMM (cublas)(row-col, A and B are in row-major) PASS!

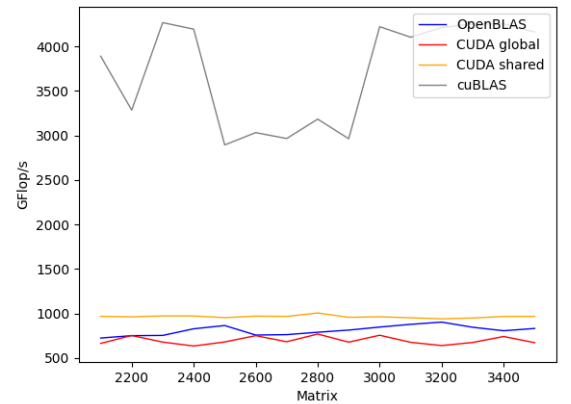
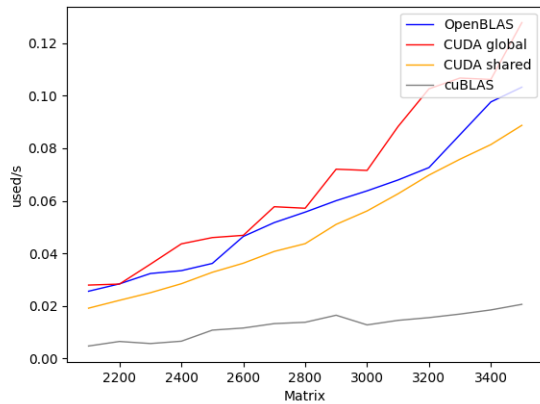
GEMM (cuda_yours)(row-col, A and B are in row-major)) used 0.00034 s for 10 bench(s) in average, 5852.01 GFlop/s
GEMM (cuda_yours)(row-col, A and B are in row-major) NOT PASS!
```



As you can see, the cuBLAS is the fastest, than is OpenBLAS, CUDA using shared memory, CUDA using global memory and basic code using OpenMP.

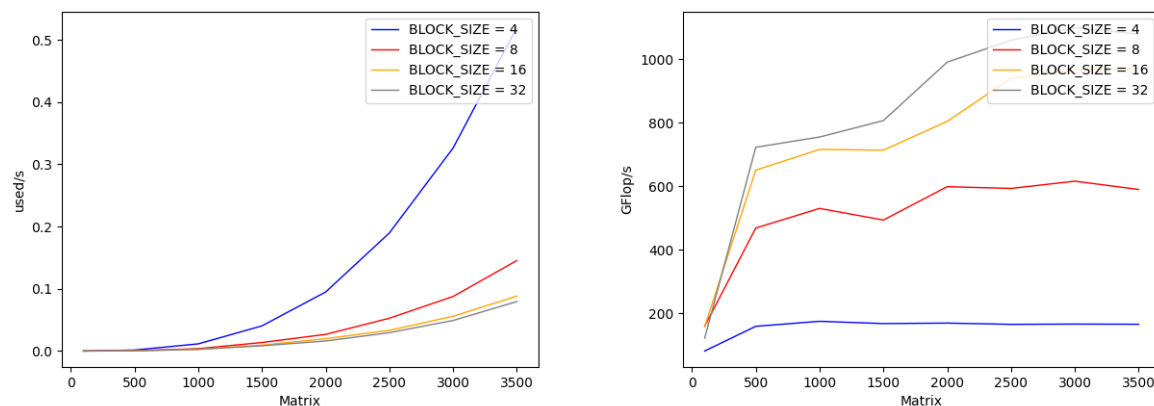
For better comparison, I use the given remote device to run the code(the Matrix from 2100 to 3500).

Here is the result:



The CUDA GEMM code optimized by yourself with your own techniques...

In theory, using more blocks is better, but cost more memory. I try the BLOCK\_SIZE 4, 8, 16, 32 to run the code, here is the performance:



The BLOCK\_SIZE for 32 is the faster. Although the increase is not obvious compared to 32, but as the width of Matrix increases, it will do more. And I try the BLOCK\_SIZE for 64, but it doesn't work, so a block have up to 1024 ( $32 \times 32$ ) threads (or more) in the device RTX 2060 SUPER?

That's all

End!