M2-BIG DATA
GPGPU - Chapter 13

Exercice 2

Réalisé par:
Gaby Maroun

Encadré par:
Dr. Etancelin JM

March 8, 2021

Objectives

Parallelize a matrix-matrix multiplication algorithm using OpenACC.

Instructions

From the given code (host sequential matrix multiplication) write an OpenACC version with explicit data management. Write 3 versions :

1. Naive version with only parallel and loop directives. Make sure that you take into account all informations given by the compiler. *Solution can be found in the file 1-basicMatMul.cxx*
2. Version with enhanced description of the algoritm collapse or tile. *Solution can be found in the file 2-basicMatMul.cxx*
3. Optimal version with both description of the algorithm and association to OpenACC levels of parallelism. *Solution can be found in the file 3-basicMatMul.cxx*

Questions

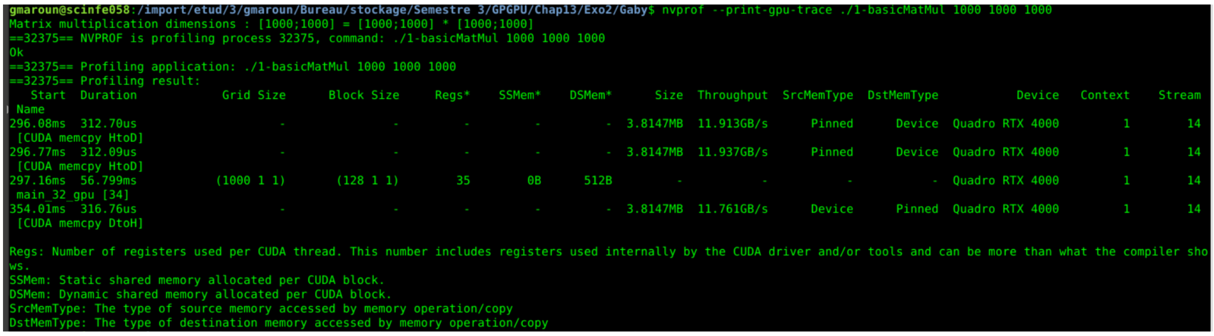
1. Explain all your choices of optimisation in version 2 and 3.

   In version 2, $Collapsed$ was used because it takes the next n(here 2) tightly-nested loops, folds them into one, what we call flattened loop and applies the OpenACC directives to the new loop.

   In version 3, $gang$ was used because each gang executes same code sequentially and independently and have 1 or more workers and share resources(such as cache, the streaming multiprocessor, etc.). That will allow us to use the vector for the multiplication loop and optimize the work.

2. Compare all the 5 versions of matrix multiplication you wrote so far : 2 CUDA versions from chapter 4 and 3 from this exercice. What is the best version ? EXPLAIN

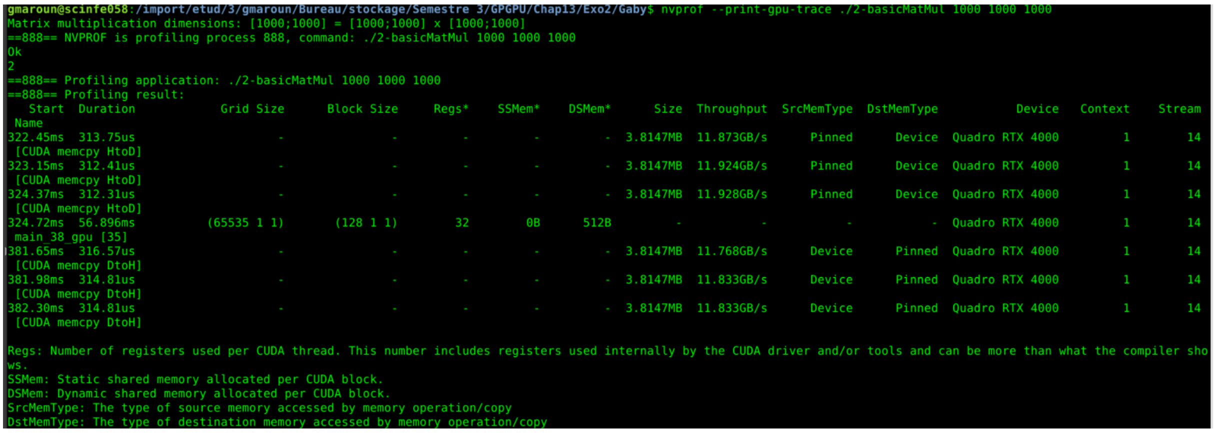
The Naive version of the code:



```
gmaroun@scinfe058:/import/etud/3/gmaroun/Bureau/stockage/Semestre 3/GPGPU/Chap13/Exo2/Gaby$ nvprof --print-gpu-trace ./1-basicMatMul 1000 1000 1000
Matrix multiplication dimensions : [1000;1000] = [1000;1000] * [1000;1000]
==32375== NVPROF is profiling process 32375, command: ./1-basicMatMul 1000 1000 1000
Ok
==32375== Profiling application: ./1-basicMatMul 1000 1000 1000
==32375== Profiling result:
   Start  Duration            Grid Size      Block Size     Regs*    SSMem*    DSMem*      Size  Throughput  SrcMemType  DstMemType           Device   Context    Stream
 Name
296.08ms  312.70us                    -               -         -         -         -  3.8147MB  11.913GB/s      Pinned      Device  Quadro RTX 4000         1        14
 [CUDA memcpy HtoD]
296.77ms  312.09us                    -               -         -         -         -  3.8147MB  11.937GB/s      Pinned      Device  Quadro RTX 4000         1        14
 [CUDA memcpy HtoD]
297.16ms  56.799ms          (1000 1 1)       (128 1 1)        35        0B      512B         -           -           -           -  Quadro RTX 4000         1        14
 main_32_gpu [34]
354.01ms  316.76us                    -               -         -         -         -  3.8147MB  11.761GB/s      Device      Pinned  Quadro RTX 4000         1        14
 [CUDA memcpy DtoH]

Regs: Number of registers used per CUDA thread. This number includes registers used internally by the CUDA driver and/or tools and can be more than what the compiler sho
ws.
SSMem: Static shared memory allocated per CUDA block.
DSMem: Dynamic shared memory allocated per CUDA block.
SrcMemType: The type of source memory accessed by memory operation/copy
DstMemType: The type of destination memory accessed by memory operation/copy
```

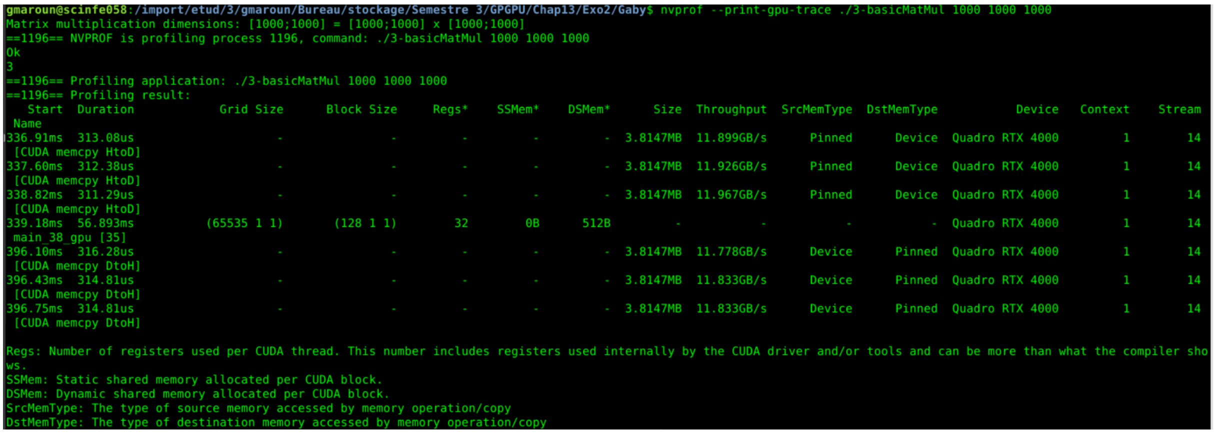
Version with enhanced description of the algoritm collapse:



```
gmaroun@scinfe058:/import/etud/3/gmaroun/Bureau/stockage/Semestre 3/GPGPU/Chap13/Exo2/Gaby$ nvprof --print-gpu-trace ./2-basicMatMul 1000 1000 1000
Matrix multiplication dimensions: [1000;1000] = [1000;1000] x [1000;1000]
==888== NVPROF is profiling process 888, command: ./2-basicMatMul 1000 1000 1000
Ok
2
==888== Profiling application: ./2-basicMatMul 1000 1000 1000
==888== Profiling result:
   Start  Duration            Grid Size      Block Size     Regs*    SSMem*    DSMem*      Size  Throughput  SrcMemType  DstMemType           Device   Context    Stream
 Name
322.45ms  313.75us                    -               -         -         -         -  3.8147MB  11.873GB/s      Pinned      Device  Quadro RTX 4000         1        14
 [CUDA memcpy HtoD]
323.15ms  312.41us                    -               -         -         -         -  3.8147MB  11.924GB/s      Pinned      Device  Quadro RTX 4000         1        14
 [CUDA memcpy HtoD]
324.37ms  312.31us                    -               -         -         -         -  3.8147MB  11.928GB/s      Pinned      Device  Quadro RTX 4000         1        14
 [CUDA memcpy HtoD]
324.72ms  56.896ms          (65535 1 1)       (128 1 1)        32        0B      512B         -           -           -           -  Quadro RTX 4000         1        14
 main_38_gpu [35]
381.65ms  316.57us                    -               -         -         -         -  3.8147MB  11.768GB/s      Device      Pinned  Quadro RTX 4000         1        14
 [CUDA memcpy DtoH]
381.98ms  314.81us                    -               -         -         -         -  3.8147MB  11.833GB/s      Device      Pinned  Quadro RTX 4000         1        14
 [CUDA memcpy DtoH]
382.30ms  314.81us                    -               -         -         -         -  3.8147MB  11.833GB/s      Device      Pinned  Quadro RTX 4000         1        14
 [CUDA memcpy DtoH]

Regs: Number of registers used per CUDA thread. This number includes registers used internally by the CUDA driver and/or tools and can be more than what the compiler sho
ws.
SSMem: Static shared memory allocated per CUDA block.
DSMem: Dynamic shared memory allocated per CUDA block.
SrcMemType: The type of source memory accessed by memory operation/copy
DstMemType: The type of destination memory accessed by memory operation/copy
```

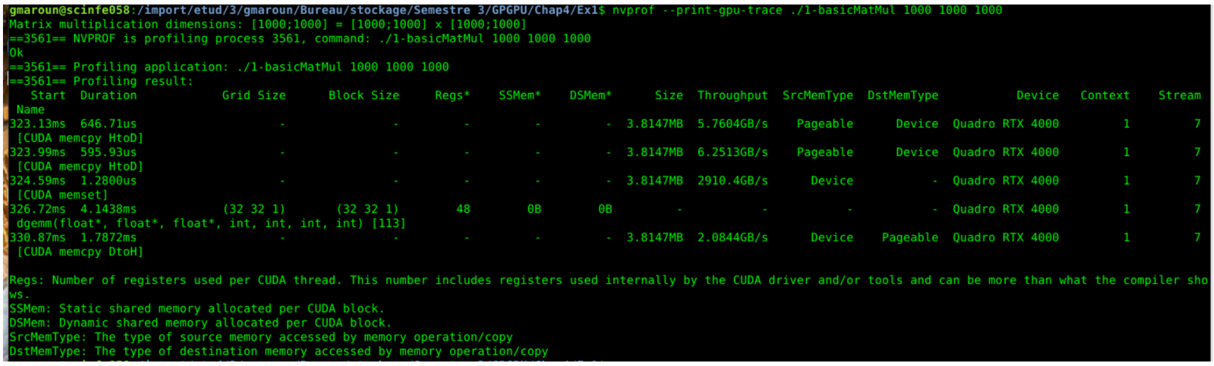
Optimal version with both description of the algorithm and association to OpenACC levels of parallelism



```
gmaroun@scinfe058:/import/etud/3/gmaroun/Bureau/stockage/Semestre 3/GPGPU/Chap13/Exo2/Gaby$ nvprof --print-gpu-trace ./3-basicMatMul 1000 1000 1000
Matrix multiplication dimensions: [1000;1000] = [1000;1000] x [1000;1000]
==1196== NVPROF is profiling process 1196, command: ./3-basicMatMul 1000 1000 1000
Ok
3
==1196== Profiling application: ./3-basicMatMul 1000 1000 1000
==1196== Profiling result:
   Start  Duration            Grid Size      Block Size     Regs*    SSMem*    DSMem*      Size  Throughput  SrcMemType  DstMemType           Device   Context    Stream
 Name
336.91ms  313.08us                    -               -         -         -         -  3.8147MB  11.899GB/s      Pinned      Device  Quadro RTX 4000         1        14
 [CUDA memcpy HtoD]
337.60ms  312.38us                    -               -         -         -         -  3.8147MB  11.926GB/s      Pinned      Device  Quadro RTX 4000         1        14
 [CUDA memcpy HtoD]
338.82ms  311.29us                    -               -         -         -         -  3.8147MB  11.967GB/s      Pinned      Device  Quadro RTX 4000         1        14
 [CUDA memcpy HtoD]
339.18ms  56.893ms          (65535 1 1)       (128 1 1)        32        0B      512B         -           -           -           -  Quadro RTX 4000         1        14
 main_38_gpu [35]
396.10ms  316.28us                    -               -         -         -         -  3.8147MB  11.778GB/s      Device      Pinned  Quadro RTX 4000         1        14
 [CUDA memcpy DtoH]
396.43ms  314.81us                    -               -         -         -         -  3.8147MB  11.833GB/s      Device      Pinned  Quadro RTX 4000         1        14
 [CUDA memcpy DtoH]
396.75ms  314.81us                    -               -         -         -         -  3.8147MB  11.833GB/s      Device      Pinned  Quadro RTX 4000         1        14
 [CUDA memcpy DtoH]

Regs: Number of registers used per CUDA thread. This number includes registers used internally by the CUDA driver and/or tools and can be more than what the compiler sho
ws.
SSMem: Static shared memory allocated per CUDA block.
DSMem: Dynamic shared memory allocated per CUDA block.
SrcMemType: The type of source memory accessed by memory operation/copy
DstMemType: The type of destination memory accessed by memory operation/copy
```

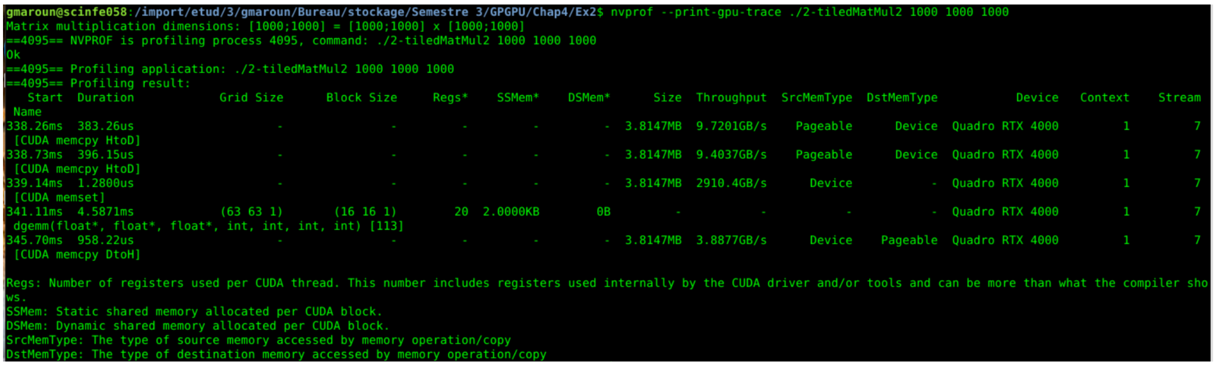
The CUDA version from chapter 4 exercice 1:



```
gmaroun@scinfe058:/import/etud/3/gmaroun/Bureau/stockage/Semestre 3/GPGPU/Chap4/Ex1$ nvprof --print-gpu-trace ./1-basicMatMul 1000 1000 1000
Matrix multiplication dimensions: [1000;1000] = [1000;1000] x [1000;1000]
==3561== NVPROF is profiling process 3561, command: ./1-basicMatMul 1000 1000 1000
Ok
==3561== Profiling application: ./1-basicMatMul 1000 1000 1000
==3561== Profiling result:
   Start  Duration            Grid Size      Block Size     Regs*    SSMem*    DSMem*      Size  Throughput  SrcMemType  DstMemType           Device   Context    Stream
 Name
323.13ms  646.71us                    -               -         -         -         -  3.8147MB  5.7604GB/s    Pageable      Device  Quadro RTX 4000         1         7
 [CUDA memcpy HtoD]
323.99ms  595.93us                    -               -         -         -         -  3.8147MB  6.2513GB/s    Pageable      Device  Quadro RTX 4000         1         7
 [CUDA memcpy HtoD]
324.59ms  1.2800us                    -               -         -         -         -  3.8147MB  2910.4GB/s      Device           -  Quadro RTX 4000         1         7
 [CUDA memset]
326.72ms  4.1438ms           (32 32 1)       (32 32 1)        48        0B        0B         -           -           -           -  Quadro RTX 4000         1         7
 dgemm(float*, float*, float*, int, int, int, int) [113]
330.87ms  1.7872ms                    -               -         -         -         -  3.8147MB  2.0844GB/s      Device    Pageable  Quadro RTX 4000         1         7
 [CUDA memcpy DtoH]

Regs: Number of registers used per CUDA thread. This number includes registers used internally by the CUDA driver and/or tools and can be more than what the compiler sho
ws.
SSMem: Static shared memory allocated per CUDA block.
DSMem: Dynamic shared memory allocated per CUDA block.
SrcMemType: The type of source memory accessed by memory operation/copy
DstMemType: The type of destination memory accessed by memory operation/copy
```

The CUDA tiled version from chapter 4 exercice 2:



```
gmaroun@scinfe058:/import/etud/3/gmaroun/Bureau/stockage/Semestre 3/GPGPU/Chap4/Ex2$ nvprof --print-gpu-trace ./2-tiledMatMul2 1000 1000 1000
Matrix multiplication dimensions: [1000;1000] = [1000;1000] x [1000;1000]
==4095== NVPROF is profiling process 4095, command: ./2-tiledMatMul2 1000 1000 1000
Ok
==4095== Profiling application: ./2-tiledMatMul2 1000 1000 1000
==4095== Profiling result:
   Start  Duration            Grid Size      Block Size     Regs*    SSMem*    DSMem*      Size  Throughput  SrcMemType  DstMemType           Device   Context    Stream
 Name
338.26ms  383.26us                    -               -         -         -         -  3.8147MB  9.7201GB/s    Pageable      Device  Quadro RTX 4000         1         7
 [CUDA memcpy HtoD]
338.73ms  396.15us                    -               -         -         -         -  3.8147MB  9.4037GB/s    Pageable      Device  Quadro RTX 4000         1         7
 [CUDA memcpy HtoD]
339.14ms  1.2800us                    -               -         -         -         -  3.8147MB  2910.4GB/s      Device           -  Quadro RTX 4000         1         7
 [CUDA memset]
341.11ms  4.5871ms            (63 63 1)       (16 16 1)        20  2.0000KB        0B         -           -           -           -  Quadro RTX 4000         1         7
 dgemm(float*, float*, float*, int, int, int, int) [113]
345.70ms  958.22us                    -               -         -         -         -  3.8147MB  3.8877GB/s      Device    Pageable  Quadro RTX 4000         1         7
 [CUDA memcpy DtoH]

Regs: Number of registers used per CUDA thread. This number includes registers used internally by the CUDA driver and/or tools and can be more than what the compiler sho
ws.
SSMem: Static shared memory allocated per CUDA block.
DSMem: Dynamic shared memory allocated per CUDA block.
SrcMemType: The type of source memory accessed by memory operation/copy
DstMemType: The type of destination memory accessed by memory operation/copy
```

We can see a slight advance in speed from the openACC based codes, and that could be for it's specialization in working on the accelerator

La fin.