Université de Pau et des Pays de l'Adour

Exercice 1 Chapitre 13

Master 2 Big Data

GPGPU

Realisé par :

Manon Bédère

L'objectif de cet exercice est de développer deux codes permettant d'additionner deux vecteurs en utilisant la méthode OpenACC.

Questions

1) Use the profiler to compare the 3 versions of vector add : CUDA, OpenACC with generated data transfers and OpenACC explicit data managment. Are the execution time of the whole algorithm (transfers + computations) are almost the same ? Explain ?

Les temps ci-dessous ont été observés pour des matrices de taille 1000*1000.

Le temps d'éxécution obtenu avec la version cuda de l'exercice 1 du chapitre 4 est la suivante :

```
mbedere@scinfe051:/import/etud/26/mbedere/Bureau/gpgpu/Chapitre 4 exercice 1$ ./1-basicMatMul 1000 1000 1000
Matrix multiplication dimensions: [1000;1000] = [1000;1000] x [1000;1000]
Ok
mbedere@scinfe051:/import/etud/26/mbedere/Bureau/gpgpu/Chapitre 4 exercice 1$ nvprof --print-gpu-trace ./1-basicMatMul 1000 1000 1000
Matrix multiplication dimensions: [1000;1000] = [1000;1000] x [1000;1000]
==1367== NVPROF is profiling process 1367, command: ./1-basicMatMul 1000 1000 1000
Ok
==1367== Profiling application: ./1-basicMatMul 1000 1000 1000
==1367== Profiling result:
   Start  Duration            Grid Size      Block Size      Regs*    SSMem*    DSMem*      Size  Throughput  SrcMemType  DstMemType           Device   Context   Stream
   Name
288.46ms  382.77us                   -               -          -         -         -  3.8147MB  9.7324GB/s    Pageable      Device  Quadro RTX 4000         1        7
  [CUDA memcpy HtoD]
288.93ms  366.84us                   -               -          -         -         -  3.8147MB  10.155GB/s    Pageable      Device  Quadro RTX 4000         1        7
  [CUDA memcpy HtoD]
289.31ms  1.2800us                   -               -          -         -         -  3.8147MB  2910.4GB/s      Device           -  Quadro RTX 4000         1        7
  [CUDA memset]
289.32ms  4.1324ms         (32 32 1)       (32 32 1)         48        0B        0B         -           -           -           -  Quadro RTX 4000         1        7
  dgemm(float*, float*, float*, int, int, int, int) [113]
293.45ms  922.15us                   -               -          -         -         -  3.8147MB  4.0398GB/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 shows.
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
```

Maintenant, avec le code OpenACC (code : addvector.c), on obtient les temps suivants :

```
mbedere@scinfe051:/import/etud/26/mbedere/Bureau/gpgpu/Chapitre13exercice1$ nvc -o vectoradd1 vectoradd1.c
mbedere@scinfe051:/import/etud/26/mbedere/Bureau/gpgpu/Chapitre13exercice1$ nvc -o vectoradd vectoradd.c
mbedere@scinfe051:/import/etud/26/mbedere/Bureau/gpgpu/Chapitre13exercice1$ nvc -acc=gpu -Minfo=accel -o  vectoradd vectoradd.c
main:
     24, Generating copyin(a[:1000]) [if not already present]
         Generating copyout(c[:1000]) [if not already present]
         Generating copyin(b[:1000]) [if not already present]
mbedere@scinfe051:/import/etud/26/mbedere/Bureau/gpgpu/Chapitre13exercice1$ nvprof --print-gpu-trace ./vectoradd
Vector size is 1000
==29367== NVPROF is profiling process 29367, command: ./vectoradd
final result:0.000000
==29367== Profiling application: ./vectoradd
==29367== Profiling result:
   Start  Duration            Grid Size      Block Size      Regs*    SSMem*    DSMem*      Size  Throughput  SrcMemType  DstMemType           Device   Context   Stream
   Name
298.76ms  1.6320us                   -               -          -         -         -  3.9063KB  2.2827GB/s      Pinned      Device  Quadro RTX 4000         1       14
  [CUDA memcpy HtoD]
298.79ms  1.2800us                   -               -          -         -         -  3.9063KB  2.9104GB/s      Pinned      Device  Quadro RTX 4000         1       14
  [CUDA memcpy HtoD]
298.83ms  2.6560us                   -               -          -         -         -  3.9063KB  1.4026GB/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 shows.
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
```

Avec le code (addvector1.c), on obtient les temps suivants:

```
mbedere@scinfe051:/import/etud/26/mbedere/Bureau/gpgpu/Chapitre13exercice1$ nvc -acc=gpu -Minfo=accel -o  vectoradd1 vectoradd1.c
main:
     27, Generating copyin(a[:1000]) [if not already present]
         Generating copyout(c[:1000]) [if not already present]
         Generating copyin(b[:1000]) [if not already present]
         Generating Tesla code
         27, #pragma acc loop gang, vector(128) /* blockIdx.x threadIdx.x */
mbedere@scinfe051:/import/etud/26/mbedere/Bureau/gpgpu/Chapitre13exercice1$ nvprof --print-gpu-trace ./vectoradd1
Vector size is 1000
==29947== NVPROF is profiling process 29947, command: ./vectoradd1
==29947== Profiling application: ./vectoradd1
==29947== Profiling result:
   Start  Duration         Grid Size      Block Size    Regs*   SSMem*    DSMem*     Size  Throughput  SrcMemType  DstMemType           Device  Context  Strea
 Name
311.91ms  1.7590us               -              -        -        -        -   3.9063KB  2.1178GB/s     Pinned      Device  Quadro RTX 4000        1        1
 [CUDA memcpy HtoD]
311.94ms  1.2790us               -              -        -        -        -   3.9063KB  2.9127GB/s     Pinned      Device  Quadro RTX 4000        1        1
 [CUDA memcpy HtoD]
311.99ms  1.9200us          (8 1 1)      (128 1 1)       16       0B       0B        -          -           -           -  Quadro RTX 4000        1        1
 main_27_gpu [34]
312.02ms  2.5280us               -              -        -        -        -   3.9063KB  1.4736GB/s     Device      Pinned  Quadro RTX 4000        1        1
 [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 s
ows.
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
```

Les temps observés sont quaisment simillaire même si on observe des temps plus longs pour le code du chapitre 4. Cela peut s'expliquer du fait que OpenACC calque sur OpenMP mais aussi car OpenACC est spécifique pour les accélérateurs.