Tecnológico de Monterrey

Campus Santa Fe

Programación Multinúcleo

Assignment 3

By Luis Carlos Arias Camacho

Objective.

The purpose of this assignment is to develop a software that performs a matrix multiplication with Tilling on shared memory with the next implementations

- Sequentially on CPU.
- Parallel on GPU.
- Parallel on GPU with Tilling.

With this information we are going to be able to learn and comprehend some of the basics of multithreading in CPU and GPU.

Test Specifications

All the test were made on the server that the professor gave us access to test our programs, that is because I do not have a Laptop with an Nvidia GPU (MacBook Pro 2012).

Also in the assignment there was specified that we should use a 8x8, 16x16 and 32x32 Tile matrices with a Matrix Multiplication of two Matrices of N X N (N = 2000)

Files in the Assignment and Functionality

1. **common.h**
   This file just defines the SAFE CALLS for the device call in GPU

2. **custom.h**
   Other files to fill the data in the matrices and to check them

3. **Assignment3.cu**
   This file has the complete implementation.

   To compile use:

   ```
   make
   ```

Results:

16X16 Tilling times on ms.		
CPU	**GPU(128 threads)**	**Tilled GPU (16X16)**
74712.18	1372.77	175.45
75101.36	1352.17	176.69
74653.54	1343.23	179.33
74826.48	1374.45	176.49
74564.91	1363.96	180.96
74622.48	1376.94	178.77
74877.43	1345.31	177.20
74556.99	1361.24	175.81
74742.45	1369.99	179.08
74962.61	1371.54	176.98
75081.76	1358.95	180.86
75132.67	1378.80	174.13
75703.16	1346.11	180.08
75424.11	1349.65	174.48
75449.55	1362.10	177.64
74802.92	1371.01	176.90
74876.34	1369.93	184.82
75205.35	1358.44	175.02
75030.36	1382.12	176.70
76176.79	1345.35	179.46

Average.

CPU: 75453.08 GPU: 1357.47 TIilled: 178.24

Speedups:

CPU / GPU: 55.58 CPU / Tilled GPU (8x8): 423.32 GPU / Tilled GPU (8x8): 7.62

8X8 Tilling times on ms.		
CPU	**GPU(256 threads)**	**Tilled GPU (8X8)**
74712.18	718.71	237.94
75101.36	749.11	238.06
74653.54	688.46	228.40
74826.48	759.20	227.73
74564.91	745.43	239.52
74622.48	733.21	240.34
74877.43	686.07	230.20
74556.99	686.19	239.57
74742.45	687.32	231.34
74962.61	735.19	239.28
75081.76	684.96	241.06
75132.67	684.92	229.39
75703.16	698.81	228.01
75424.11	684.96	238.13
75449.55	735.71	237.89
74802.92	685.20	240.52
74876.34	692.52	233.50
75205.35	729.06	238.37
75030.36	685.08	242.41
76176.79	689.33	243.37

Average.

CPU: 75453.08 GPU: 705.72 Tlilled: 238.02

Speedups:

CPU / GPU: 106.91 CPU / Tilled GPU (16x16): 317 GPU / Tilled GPU (16x16): 2.96

32X32 Tilling times on ms.		
CPU	**GPU(64 threads)**	**Tilled GPU (32X32)**
74712.18	2461.45	208.42
75101.36	2468.00	209.26
74653.54	2483.13	207.32
74826.48	2395.01	207.28
74564.91	2394.88	209.38
74622.48	2474.93	207.22
74877.43	2394.18	207.25
74556.99	2475.14	207.64
74742.45	2401.01	207.38
74962.61	2457.03	210.07
75081.76	2414.17	211.18
75132.67	2402.82	210.07
75703.16	2489.71	210.99
75424.11	2407.51	212.14
75449.55	2409.48	209.94
74802.92	2484.15	208.72
74876.34	2413.53	211.19
75205.35	2398.33	211.19
75030.36	2468.38	209.92
76176.79	2402.52	209.75

Average.

CPU: 75453.08 GPU: 2408.12 TIilled: 209.41

Speedups:

CPU / GPU: 31.33 CPU / Tilled GPU (32x32): 360.31 GPU / Tilled GPU (16x16): 11.5

Conclusion.

We can see that an increase of the shared memory tile really speeds up the performance of our program, because it makes less calls to the main memory. It can perform better with a shared memory of 256 spaces.

Also we can observe that the performance on a normal GPU increases with a number of 256 threads per block.