Tampere University

COMP.CE.350

Multicore and GPU Programming Lab Work Autumn 2020

October 29, 2020

Student's name : Trinh Gia Huy

Student's number: H290290

Student's email: giahuy.trinh@tuni.fi

I. 6.1

-no optimization

```
[bdgitr@linux-desktop4 Project]$ gcc -o parallel parallel.c -std=c99 -lglut -lGL -lm
[bdgitr@linux-desktop4 Project]$ ./parallel
Error check passed!
Total frametime: 5249ms, satelite moving: 582ms, space coloring: 2276ms.
Total frametime: 2585ms, satelite moving: 302ms, space coloring: 2238ms.
Total frametime: 2611ms, satelite moving: 292ms, space coloring: 2287ms.
Total frametime: 2839ms, satelite moving: 292ms, space coloring: 2514ms.
Total frametime: 2708ms, satelite moving: 349ms, space coloring: 2270ms.
Total frametime: 2820ms, satelite moving: 348ms, space coloring: 2438ms.
```

-most optimizations, no vectorization

```
[bdgitr@linux-desktop4 Project]$ gcc -o parallel parallel.c -std=c99 -lglut -lGL -lm -O2
[bdgitr@linux-desktop4 Project]$ ./parallel
Error check passed!
Total frametime: 3012ms, satelite moving: 568ms, space coloring: 1150ms.
Total frametime: 1474ms, satelite moving: 293ms, space coloring: 1137ms.
Total frametime: 1507ms, satelite moving: 282ms, space coloring: 1188ms.
Total frametime: 1457ms, satelite moving: 283ms, space coloring: 1147ms.
Total frametime: 1450ms, satelite moving: 297ms, space coloring: 1121ms.
Total frametime: 1429ms, satelite moving: 282ms, space coloring: 1118ms.
```

Total frametime: 1432ms, satelite moving: 283ms, space coloring: 1117ms.
Total frametime: 1429ms, satelite moving: 282ms, space coloring: 1116ms.

-ftree-vectorize

```
[bdgitr@linux-desktop4 Project]$ gcc -o parallel parallel.c -std=c99 -lglut -lGL -lm -O2 -ftree-vectorize -fopt-info-vec
parallel.c:183:4: optimized: loop vectorized using 16 byte vectors
parallel.c:394:4: optimized: loop vectorized using 16 byte vectors
parallel.c:491:4: optimized: basic block part vectorized using 16 byte vectors
[bdgitr@linux-desktop4 Project]$ ./parallel
Error check passed!
Total frametime: 3721ms, satelite moving: 691ms, space coloring: 1333ms.
Total frametime: 1692ms, satelite moving: 353ms, space coloring: 1275ms.
Total frametime: 1533ms, satelite moving: 311ms, space coloring: 1156ms.
Total frametime: 1509ms, satelite moving: 321ms, space coloring: 1141ms.
Total frametime: 1513ms, satelite moving: 287ms, space coloring: 1124ms.
Total frametime: 1466ms, satelite moving: 284ms, space coloring: 1148ms.
Total frametime: 1446ms, satelite moving: 283ms, space coloring: 1132ms.
```

It informs about 2 loop 183:4 and 394:4 are able to vectorize with 16 byte-vector.

4.

*The -ffast-math give the best performance. Since it has vectorized up to 4 loops. The -ffast-math flags enables a range of optimizations that provide faster, though sometimes less precise, mathematical operations.

```
[bdgitr@linux-desktop6 Project]$ gcc -o parallel parallel.c -std=c99 -lglut -lGL -lm -O2 -ftree-vectorize -fopt-info-vec -ffast-math
parallel.c:183:4: optimized: loop vectorized using 16 byte vectors
parallel.c:149:7: optimized: loop vectorized using 16 byte vectors
parallel.c:394:4: optimized: loop vectorized using 16 byte vectors
parallel.c:359:7: optimized: loop vectorized using 16 byte vectors
parallel.c:491:4: optimized: basic block part vectorized using 16 byte vectors
[bdgitr@linux-desktop6 Project]$ ./parallel
Error check passed!
Total frametime: 1797ms, satelite moving: 210ms, space coloring: 736ms.
Total frametime: 883ms, satelite moving: 105ms, space coloring: 738ms.
Total frametime: 867ms, satelite moving: 106ms, space coloring: 731ms.
Total frametime: 862ms, satelite moving: 107ms, space coloring: 729ms.
Total frametime: 883ms, satelite moving: 105ms, space coloring: 751ms.
Total frametime: 942ms, satelite moving: 107ms, space coloring: 806ms.
Total frametime: 905ms, satelite moving: 119ms, space coloring: 753ms.
Total frametime: 875ms, satelite moving: 112ms, space coloring: 735ms.
```

Total frametime: 888ms, satelite moving: 109ms, space coloring: 749ms.

Other flags:

-fopenmp

The -fopenmp flags enable handling of OpenMP directives and generate parallel code.

```
[bdgitr@linux-desktop4 Project]$ gcc -o parallel parallel.c -std=c99 -lglut -lGL -lm -O2 -ftree-vectorize -fopt-info-vec -fopenmp
parallel.c:183:4: optimized: loop vectorized using 16 byte vectors
parallel.c:394:4: optimized: loop vectorized using 16 byte vectors
parallel.c:491:4: optimized: basic block part vectorized using 16 byte vectors
[bdgitr@linux-desktop4 Project]$ ./parallel
Error check passed!
Total frametime: 3060ms, satelite moving: 586ms, space coloring: 1148ms.
Total frametime: 1486ms, satelite moving: 284ms, space coloring: 1159ms.
Total frametime: 1467ms, satelite moving: 312ms, space coloring: 1124ms.
Total frametime: 1456ms, satelite moving: 285ms, space coloring: 1140ms.
Total frametime: 1471ms, satelite moving: 307ms, space coloring: 1127ms.
Total frametime: 1478ms, satelite moving: 302ms, space coloring: 1136ms.
Total frametime: 1465ms, satelite moving: 297ms, space coloring: 1131ms.
```

-mpc32

```
[bdgitr@linux-desktop6 Project]$ gcc -o parallel parallel.c -std=c99 -lglut -lGL -lm -O2 -ftree-vectorize -fopt-info-vec -mpc32
parallel.c:183:4: optimized: loop vectorized using 16 byte vectors
parallel.c:394:4: optimized: loop vectorized using 16 byte vectors
parallel.c:491:4: optimized: basic block part vectorized using 16 byte vectors
bdgitr@linux-desktop6 Project]$ ./parallel
Error check passed!
Total frametime: 2951ms, satelite moving: 572ms, space coloring: 1137ms.
Total frametime: 1442ms, satelite moving: 282ms, space coloring: 1118ms.
Total frametime: 1442ms, satelite moving: 282ms, space coloring: 1131ms.
Total frametime: 1440ms, satelite moving: 282ms, space coloring: 1116ms.
Total frametime: 1446ms, satelite moving: 288ms, space coloring: 1123ms.
Total frametime: 1448ms, satelite moving: 287ms, space coloring: 1120ms.
Total frametime: 1448ms, satelite moving: 282ms, space coloring: 1123ms.
```

-mpc64

```
[bdgitr@linux-desktop6 Project]$ gcc -o parallel parallel.c -std=c99 -lglut -lGL -lm -O2 -ftree-vectorize -fopt-info-vec -mpc64
parallel.c:183:4: optimized: loop vectorized using 16 byte vectors
parallel.c:394:4: optimized: loop vectorized using 16 byte vectors
```

parallel.c:491:4: optimized: basic block part vectorized using 16 byte vectors
[bdgitr@linux-desktop6 Project]$ ./parallel
Error check passed!
Total frametime: 2991ms, satelite moving: 574ms, space coloring: 1156ms.
Total frametime: 1476ms, satelite moving: 308ms, space coloring: 1132ms.
Total frametime: 1444ms, satelite moving: 288ms, space coloring: 1123ms.
Total frametime: 1432ms, satelite moving: 282ms, space coloring: 1118ms.
Total frametime: 1443ms, satelite moving: 286ms, space coloring: 1124ms.
Total frametime: 1446ms, satelite moving: 284ms, space coloring: 1122ms.
Total frametime: 1436ms, satelite moving: 281ms, space coloring: 1119ms

-mhard-float
parallel.c:183:4: optimized: loop vectorized using 16 byte vectors
parallel.c:394:4: optimized: loop vectorized using 16 byte vectors
parallel.c:491:4: optimized: basic block part vectorized using 16 byte vectors
[bdgitr@linux-desktop6 Project]$ ./parallel
Error check passed!
Total frametime: 3015ms, satelite moving: 569ms, space coloring: 1187ms.
Total frametime: 1523ms, satelite moving: 282ms, space coloring: 1198ms.
Total frametime: 1448ms, satelite moving: 289ms, space coloring: 1125ms.
Total frametime: 1436ms, satelite moving: 287ms, space coloring: 1122ms.
Total frametime: 1456ms, satelite moving: 286ms, space coloring: 1139ms.
Total frametime: 1436ms, satelite moving: 281ms, space coloring: 1124ms.
Total frametime: 1439ms, satelite moving: 283ms, space coloring: 1117ms.
Total frametime: 1694ms, satelite moving: 290ms, space coloring: 1374ms.

-msoft-float
parallel.c:183:4: optimized: loop vectorized using 16 byte vectors
parallel.c:394:4: optimized: loop vectorized using 16 byte vectors
parallel.c:491:4: optimized: basic block part vectorized using 16 byte vectors
[bdgitr@linux-desktop6 Project]$ ./parallel
Error check passed!
Total frametime: 3023ms, satelite moving: 580ms, space coloring: 1174ms.
Total frametime: 1547ms, satelite moving: 303ms, space coloring: 1194ms.
Total frametime: 1457ms, satelite moving: 302ms, space coloring: 1119ms.
Total frametime: 1443ms, satelite moving: 281ms, space coloring: 1133ms.
Total frametime: 1438ms, satelite moving: 282ms, space coloring: 1116ms.
Total frametime: 1436ms, satelite moving: 281ms, space coloring: 1129ms.
Total frametime: 1432ms, satelite moving: 282ms, space coloring: 1117ms.
Total frametime: 1428ms, satelite moving: 281ms, space coloring: 1116ms.
Total frametime: 1441ms, satelite moving: 283ms, space coloring: 1133ms.

-mrtd
cc1: warning: ??-mrtd?? is ignored in 64bit mode

parallel.c:183:4: optimized: loop vectorized using 16 byte vectors
parallel.c:394:4: optimized: loop vectorized using 16 byte vectors
parallel.c:491:4: optimized: basic block part vectorized using 16 byte vectors
[bdgitr@linux-desktop6 Project]$ ./parallel
Error check passed!
Total frametime: 3219ms, satelite moving: 653ms, space coloring: 1160ms.
Total frametime: 1441ms, satelite moving: 286ms, space coloring: 1122ms.
Total frametime: 1439ms, satelite moving: 283ms, space coloring: 1125ms.
Total frametime: 1427ms, satelite moving: 281ms, space coloring: 1117ms.
Total frametime: 1430ms, satelite moving: 281ms, space coloring: 1120ms.
Total frametime: 1431ms, satelite moving: 281ms, space coloring: 1119ms.
Total frametime: 1441ms, satelite moving: 283ms, space coloring: 1121ms.
Total frametime: 1439ms, satelite moving: 281ms, space coloring: 1128ms.

As can see that most of other flag have total frame time's runtimes as average about 144x ms , much larger compared to -ffast-math flag compiler.

compiler flags which cause broken code to be generated:

-mavx (32), -mavx2(32), -mavx512f(64), -mavx512cd(64), -mfma(32),-mfma4(32),-madx(16)

I think the reason for this is these flag compiler used a different number of bytes vectocs when printing out using option -fopt-info-vec and their corresponding number of byte represent as in () above.

I. 6.3

- The Physics satellite and Graphic pixel loops are able to be parallelized to multi-thread. The physics iteration however , are not beneficial to be parallelized since those variable in that loop are not independent.
- I have split the Physics Satellite and Graphic Pixel evenly into 2 parallel process that thread can execute simultaneously. Ideally, the speed should be double.
- My code has transform into 2 for loop with different index:

*for Physics satelite loop

```
for (int i = 0; i < SATELITE_COUNT/2; ++i) {..}
for (int i = SATELITE_COUNT/2; i < SATELITE_COUNT ; ++i) {..}
```

and

*for both First Graphics satelite loop and Second Graphic satellite:

```
for (int j = 0; j < SATELITE_COUNT/2; ++j) {
```

```
for (int j = SATELITE_COUNT/2; j < SATELITE_COUNT; ++j) {..}
```

*The code transform does not have any affects on vectorization since its vectorize still on 16 byte vector as previous before splitting.

II. 6.4

*I have parallelize the Graphic pixel loop using OpenMP pragmas. Its run time is faster a lot compare to -fopenmp flag only. {using #pragma omp parallel for shared(satelites) }

[bdgitr@linux-desktop4 Project]$ gcc -o parallel parallel.c -std=c99 -lglut -lGL -lm -O2 -ftree-vectorize -fopt-info-vec -fopenmp

parallel.c:186:4: optimized: loop vectorized using 16 byte vectors
parallel.c:400:4: optimized: loop vectorized using 16 byte vectors
parallel.c:497:4: optimized: basic block part vectorized using 16 byte vectors
[bdgitr@linux-desktop4 Project]$ ./parallel
Error check passed!
Total frametime: 2320ms, satelite moving: 574ms, space coloring: 463ms.
Total frametime: 799ms, satelite moving: 284ms, space coloring: 474ms.
Total frametime: 752ms, satelite moving: 288ms, space coloring: 435ms.
Total frametime: 644ms, satelite moving: 281ms, space coloring: 330ms.
Total frametime: 682ms, satelite moving: 282ms, space coloring: 369ms.
Total frametime: 680ms, satelite moving: 283ms, space coloring: 359ms.
Total frametime: 632ms, satelite moving: 283ms, space coloring: 318ms.
Total frametime: 615ms, satelite moving: 282ms, space coloring: 301ms.
Total frametime: 744ms, satelite moving: 284ms, space coloring: 436ms.
Total frametime: 696ms, satelite moving: 287ms, space coloring: 385ms.
Total frametime: 637ms, satelite moving: 284ms, space coloring: 338ms.

*Parallelize the Physics satelite loop causes slow down.

[bdgitr@linux-desktop4 Project]$ gcc -o parallel parallel.c -std=c99 -lglut -lGL -lm -O2 -ftree-vectorize -fopt-info-vec -fopenmp
parallel.c:185:4: optimized: loop vectorized using 16 byte vectors
parallel.c:150:15: optimized: basic block part vectorized using 16 byte vectors
parallel.c:396:4: optimized: loop vectorized using 16 byte vectors
parallel.c:493:4: optimized: basic block part vectorized using 16 byte vectors
[bdgitr@linux-desktop4 Project]$ ./parallel
Error check passed!

Total frametime: 3253ms, satelite moving: 796ms, space coloring: 1139ms.
Total frametime: 1774ms, satelite moving: 604ms, space coloring: 1132ms.
Total frametime: 1849ms, satelite moving: 692ms, space coloring: 1125ms.
Total frametime: 1768ms, satelite moving: 617ms, space coloring: 1126ms.
Total frametime: 2132ms, satelite moving: 975ms, space coloring: 1130ms.
Total frametime: 1708ms, satelite moving: 518ms, space coloring: 1158ms.
Total frametime: 1645ms, satelite moving: 481ms, space coloring: 1133ms.

*Parallelize Graphics satellite. loop cause break down.

```
[bdgitr@linux-desktop4 Project]$ gcc -o parallel parallel.c -std=c99 -lglut -lGL -lm -O2 -ftree-vectorize -fopt-info-vec -fopenmp
parallel.c: In function ??parallelGraphicsEngine??:
parallel.c:227:13: error: break statement used with OpenMP for loop
  227 |             break;
      |             ^~~~~
parallel.c:240:7: error: for statement expected before ??if??
  240 |       if (!hitsSatellite) {
```

*Yes, my Laptop processors have up to native 8 threads which has scaled with the performance.

*No, I do not use the TC217 computers. My CPU is Intel® CoreTM i5-9300H processor (2.4 GHz withTurbo Boost up to 4.1 GHz). The BIOS version is Insyde Corp. V1.01.