CME213 Spring 2023 Homework 2

Ines Dormoy – idormoy@stanford.edu

April 25, 2023

By turning in this assignment, I agree by the Stanford honor code and declare that all of this is my own work.

Problem 1

```
Running main() from ./googletest-main/googletest/src/gtest_main.cc
[==========] Running 1 test from 1 test suite.
[----------] Global test environment set-up.
[----------] 1 test from testQ1
[ RUN      ] testQ1.test
Parallel
Sum Even: 757361650
Sum Odd: 742539102
Time: 0.0120495
Serial
Sum Even: 757361650
Sum Odd: 742539102
Time: 0.103744
[       OK ] testQ1.test (301 ms)
[----------] 1 test from testQ1 (302 ms total)

[----------] Global test environment tear-down
[==========] 1 test from 1 test suite ran. (302 ms total)
[  PASSED  ] 1 test.
```

Problem 2

```
    Running main() from ./googletest-main/googletest/src/gtest_main.cc
[==========] Running 7 tests from 1 test suite.
[----------] Global test environment set-up.
[----------] 7 tests from testQ2
[ RUN      ] testQ2.test1
[       OK ] testQ2.test1 (25 ms)
[ RUN      ] testQ2.test2
[       OK ] testQ2.test2 (2 ms)
[ RUN      ] testQ2.test3
[       OK ] testQ2.test3 (0 ms)
[ RUN      ] testQ2.test4
[       OK ] testQ2.test4 (2 ms)
[ RUN      ] testQ2.test5
[       OK ] testQ2.test5 (6 ms)
[ RUN      ] testQ2.serialSortTest
Serial Radix Sort: PASS
stl: 0.228604
serial radix: 0.0880066
[       OK ] testQ2.serialSortTest (455 ms)
[ RUN      ] testQ2.parallelSortTest
Parallel Radix Sort: PASS
stl: 0.225698
parallel radix: 0.027922
[       OK ] testQ2.parallelSortTest (390 ms)
[----------] 7 tests from testQ2 (883 ms total)

[----------] Global test environment tear-down
[==========] 7 tests from 1 test suite ran. (883 ms total)
[  PASSED  ] 7 tests.
```

Problem 2 Q6

```
Threads Blocks / Timing
          1     2     4     8    12    16    24    32    40    48
   1  0.047 0.046 0.046 0.052 0.052 0.052 0.057 0.062 0.064 0.086
   2  0.045 0.032 0.034 0.030 0.036 0.035 0.038 0.039 0.045 0.057
   4  0.048 0.027 0.021 0.020 0.029 0.024 0.029 0.030 0.034 0.048
   8  0.059 0.026 0.018 0.014 0.018 0.016 0.021 0.022 0.026 0.041
  12  0.055 0.033 0.028 0.019 0.026 0.024 0.030 0.029 0.033 0.049
  16  0.055 0.033 0.024 0.019 0.019 0.019 0.026 0.027 0.031 0.046
  24  0.052 0.035 0.029 0.031 0.025 0.022 0.027 0.030 0.034 0.049
  32  0.053 0.052 0.037 0.025 0.027 0.023 0.035 0.026 0.032 0.049
  40  0.054 0.056 0.041 0.028 0.025 0.021 0.027 0.028 0.031 0.049
  48  0.054 0.057 0.034 0.031 0.026 0.021 0.024 0.028 0.032 0.049

```

We observe that with only one block (first column), it is useless to add threads as those threads are not used. Adding those threads is even a waste of time as we spend time creating unused threads. Therefore, the running time increases in the first column when adding threads.
In the second column, we observe that adding a second thread induces a speedup as we use 2 blocks. Again, the runtime increases as we add unused threads.
More globally, the runtime decreases along the diagonal when we add blocks and threads. However, after n_threads = 8, we don't observe more speedup (this is because of how the icme-gpu cluster is built). We still observe that the running time increases under this diagonal (n_threads >n_blocks).
Having n_blocks >n_threads does keep the runtime equivalent when the number of blocks is not too big, but increases runtime if there are too many blocks in comparison to the number of threads (we waste time in the block-threads assignments).
Overall, the best performance is observed with n_threads = n_blocks and n_threads = 8.