REPORT

Q1)

(Note: When I execute code for Q1, I got the follow values:)

With Affinity:

Performance in Gflops 0.010 Gflop/s

Performance in Gflops 0.009 Gflop/s

Performance in Gflops 0.007 Gflop/s

Performance in Gflops 0.007 Gflop/s

Variance for with affinity: 6.85417e-05

Without Affinity:

Performance in Gflops 0.011 Gflop/s

Performance in Gflops 0.010 Gflop/s

Performance in Gflops 0.010 Gflop/s

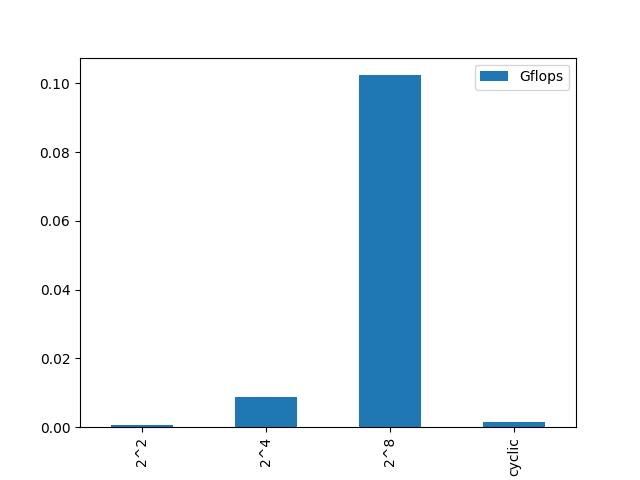
Performance in Gflops 0.010 Gflop/s

Variance for without affinity: 0.000102767

Explanation:

As observed, in case where affinity was set by us, the variance in gflops value is lesser. As we placed threads in different set of cores, we ensured that the threads were not competing for the same CPU resources. On the other hand, when we did not set affinity and it resulted in higher value for variance in gflops. This means that a lot of threads ran on a single core and they were competing each other for CPU resources which caused resource contention and hence, caused a lot of variation in performance.

Q2)



Explanation:

As observed, creating many threads that covers smaller block size is more efficient method rather than creater few threads that covers huge block size. This is due to the fact that there is effeicent use of resources when we create many threads with small block size. On the other hand, making cyclic threads also have better performace(almost as good as making many threads with block size of 4)