

ECE 459: Programming for Performance

Assignment 1

Lawrence Park

January 30, 2012

Part 2 - Benchmarking

These experiments were run on a Intel(R) Core(TM) i5-3570 CPU. It has 4 physical cores and 4 virtual CPUs.

	Time (s)
Run 1	11.041
Run 2	11.061
Run 3	11.05
Run 4	11.058
Run 5	11.06
Run 6	11.07
Average	11.06

Table 1: Benchmark results for sequential execution ($i = 800000000$)

Refer to **Table 1** and estimate runtime with 4 physical cores.

Assuming the code is 100% parallel, the speedup is calculated: $S = 1 / 1/N = 4$. Therefore, the parallel runtime is estimated to be $T_s / S = 2.765$ s.

	Time (s)
Run 1	2.938
Run 2	2.935
Run 3	2.934
Run 4	2.927
Run 5	2.927
Run 6	2.953
Average	2.94

Table 2: Benchmark results for parallel execution ($i = 800000000$, $t = 4$)

Refer to **Table 2**, does this agree with your predicted runtime? Write your answer here.

Speedup with pthread execution is calculated to be approximately 3.76 which is fairly close to our crude speedup approximation of 4. The difference can be explained by the fact that not all executed code was parallel, and there exists an overhead for creating pthreads and joining them.

Our original sequential algorithm for Monte Carlo approximation is "embarrassingly parallel," and it explains why predicted speedup is close to the actual speedup.

	Time (s)
Run 1	2.95
Run 2	2.976
Run 3	2.962
Run 4	2.97
Run 5	2.966
Run 6	2.953
Average	2.962

Table 3: Benchmark results for parallel execution ($i = 800000000$, $t = 4$)

Refer to **Table 3** calculate the speedup, and verify it is less than 4.

Intel(R) Core(TM) i5-3570 CPU does not have hyperthreading, and its virtual CPUs are of the same number as its physical cores. Speedup, therefore, is estimated at 4 (i.e. number of physical cores), and the actual speedup was 3.73.

	Time (s)
Run 1	3.2
Run 2	3.054
Run 3	3.049
Run 4	3.423
Run 5	2.961
Run 6	3.058
Average	3.124

Table 4: Benchmark results for parallel execution ($i = 800000000$, $t = 4 + 1$)

Refer to **Table 4**, calculate the speedup and compare it to **Table 3** (or **Table 2** if you don't have hyperthreading), which performs better? Write your explanation here.

The speedup with 5 threads was 3.53 which is slower than the speedup achieved with 4 threads which was 3.76. This difference makes sense because with 5 threads, the CPU cores will not have even amount of workload as they did with 4 threads. With 5 threads created, one of the 4 cores will have to execute the last thread while the other cores stay idle.