ICSI 520 Distributed & Parallel Computing - Fall 2019

Homework 3

Santhosh Ranganathan (NetID: SR582413)

sranganathan2@albany.edu

Results:

Table 1: Varying threads, fixed pushes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NUM\_THREADS (p)** | **ITERATIONS (n)** | **Execution time serial (μs)** | **Execution time parallel (μs)** | **Speedup** |
| 5 | 100 | 188.6 | 805.6 | 0.23411 |
| 10 | 100 | 188.6 | 983.4 | 0.19178 |
| 20 | 100 | 188.6 | 1012.2 | 0.18633 |
| 40 | 100 | 188.6 | 1626.2 | 0.11598 |
| 50 | 100 | 188.6 | 1905.6 | 0.09897 |
| 100 | 100 | 188.6 | 3763.8 | 0.05011 |

Table 2: Fixed threads, varying pushes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NUM\_THREADS (p)** | **ITERATIONS (n)** | **Execution time serial (μs)** | **Execution time parallel (μs)** | **Speedup** |
| 25 | 10 | 16 | 1080.5 | 0.01481 |
| 25 | 25 | 38 | 1045 | 0.03636 |
| 25 | 50 | 76.8 | 1064.6 | 0.07214 |
| 25 | 100 | 188.6 | 1086.4 | 0.17360 |
| 25 | 150 | 240 | 1578.4 | 0.15205 |
| 25 | 200 | 324.8 | 1719.8 | 0.18886 |

Table 3: Varying number of threads and varying pushes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **NUM\_THREADS (p)** | **ITERATIONS (n)** | **Execution time serial (μs)** | **Execution time parallel (μs)** | **Speedup** |
| 10 | 20 | 31.4 | 481.2 | 0.06525 |
| 20 | 40 | 67.2 | 901.6 | 0.07453 |
| 30 | 60 | 94 | 1187.2 | 0.07918 |
| 40 | 100 | 188.6 | 1730 | 0.10902 |
| 50 | 150 | 240 | 1940.8 | 0.12366 |
| 100 | 200 | 324.8 | 3693 | 0.08795 |

srun -n1 -c1 Homework3\_Serial.out 100

srun -n1 -c8 Homework3\_Parallel.out 100 10

(Averaging 5 runs)

(Parallel execution with 8 cores)

Explanation:

Parallel execution was checked for correctness by printing out the popped elements with thread numbers, checking if there are equal number of push and pop operations with stack count zero at end. Parallel execution time is consistently significantly worse than serial execution time due to thread management and using locks. Parallel execution time was still high even after removing all push and pop operations from threads. I tried to remove thread creation from timing calculation using conditional variable to create suspended threads but didn’t see much change. I did see significant improvement compared to serial when adding fake calculation and i/o to Pop() as normally operations are performed on stack elements for eg. performing i/o, DFS tree traversal etc..