Results:

Table 1: Varying threads, fixed pushes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NUM\_THREADS(p) | ITERATIONS(n) | Execution TIme Serial | Execution Time Parallel | Speedup |
| 5 | 100 | 716 | 6239 | 0.114 |
| 10 | 100 | 716 | 6636 | 0.107 |
| 20 | 100 | 716 | 6851 | 0.104 |
| 40 | 100 | 716 | 7600 | 0.094 |
| 50 | 100 | 716 | 8128 | 0.088 |
| 100 | 100 | 716 | 9637 | 0.074 |

Table 2: Fixed threads varying pushes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NUM\_THREADS (p) | ITERATIONS (n) | Execution time serial | Execution time parallel | Speedup |
| 25 | 10 | 425 | 2033 | .209 |
| 25 | 25 | 635 | 2184 | .209 |
| 25 | 50 | 765 | 2728 | .280 |
| 25 | 100 | 716 | 6119 | .117 |
| 25 | 150 | 1141 | 11061 | .103 |
| 25 | 200 | 760 | 17198 | 0.044 |

Table 3: Varying number of threads and varying pushes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NUM\_THREADS (p) | ITERATIONS (n) | Execution time serial | Execution time parallel | Speedup |
| 10 | 20 | 298 | 1295 | .230 |
| 20 | 40 | 374 | 2786 | .134 |
| 30 | 60 | 407 | 4132 | .098 |
| 40 | 100 | 716 | 6979 | .102 |
| 50 | 150 | 1141 | 13201 | .086 |
| 100 | 200 | 760 | 26498 | .028 |

Unfortunately, I did not achieve speed up. I tried two approaches. I enclosed the code for the even worse attempt at slicing the pushes. My second approach involved allowing the threads to all process items to push. If an item to push wasn’t available the thread would then start to pop items.

I suspect that the overhead of locking and unlocking could have slowed down my processing. Also, one change to the algorithm that could be tried is to change how the stack count is obtained. I currently lock and count. It might be more efficient to store it in a variable and then use that variable to give the stack count.