|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Fixed Iteration |  |  |  |  |  |
|  |  |  |  |  |  |
| Core Used | Num\_Threads(p) | ITERATIONS(n) | Execution Time Serial | Execution Time Parallel | SpeedUP |
| 5 | 5 | 100 | 594 | 665 | 0.893233 |
| 10 | 10 | 100 | 594 | 841 | 0.706302 |
| 20 | 20 | 100 | 594 | 1062 | 0.559322 |
| 20 | 40 | 100 | 594 | 2615 | 0.227151 |
| 25 | 50 | 100 | 594 | 3335 | 0.178111 |
| 30 | 100 | 100 | 594 | 4390 | 0.135308 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| Fixed Number of Threads |  |  |  |  |  |
| Core Used | Num\_Threads(p) | ITERATIONS(n) | Execution Time Serial | Execution Time Parallel | SpeedUP |
| 25 | 25 | 10 | 358 | 1205 | 0.297095 |
| 25 | 25 | 25 | 371 | 1153 | 0.321769 |
| 25 | 25 | 50 | 340 | 1290 | 0.263566 |
| 25 | 25 | 100 | 348 | 1344 | 0.258929 |
| 25 | 25 | 150 | 362 | 1354 | 0.267356 |
| 25 | 25 | 200 | 405 | 1166 | 0.347341 |
| 25 | 25 | 10000 | 1423 | 30894 | 0.046061 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| Core Used | Num\_Threads(p) | ITERATIONS(n) | Execution Time Serial | Execution Time Parallel | SpeedUP |
| 36 | 36 | 10000 | 2693 | 2620 | 1.027863 |
| 36 | 36 | 100000 | 22429 | 15632 | 1.434813 |
| 36 | 36 | 400000 | 90932 | 53692 | 1.693586 |
| 36 | 36 | 1600000 | 358373 | 219487 | 1.632776 |
| 36 | 36 | 6400000 | 1455116 | 770889 | 1.887582 |
|  |  |  |  |  |  |
| Variable Number of Thread and Variable number of Iteration |  |  |  |  |  |
| Core Used | Num\_Threads(p) | ITERATIONS(n) | Execution Time Serial | Execution Time Parallel | SpeedUP |
| 10 | 10 | 20 | 558 | 1203 | 0.46384 |
| 20 | 20 | 40 | 643 | 1872 | 0.343483 |
| 30 | 30 | 60 | 570 | 2994 | 0.190381 |
| 36 | 40 | 100 | 364 | 1834 | 0.198473 |
| 36 | 50 | 150 | 355 | 2308 | 0.153813 |
| 36 | 100 | 200 | 452 | 4686 | 0.096458 |
|  |  |  |  |  |  |

In this Parallel Programming multiple threads performs the simultaneous Push operation and a single thread performs Pop Operation. I have not used any mutex in this. I am using Array of Integer as a stack to store the elements. The “GetStackCount()” is performed after all the push and pop operation is completed, this function fetches the current number of elements in the stack which is zero at last. This function” GetStackCount()” may return a minor error value when it is called during the push and pop operation.