|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N | P | Serial execution time (microseconds) (P=1) | Parallel execution time (microseconds) | SpeedUP |
| 1000 | 2 | 38 | 251 | 0.151394 |
| 10000 | 2 | 109 | 259 | 0.420849 |
| 100000 | 2 | 640 | 569 | 1.12478 |
| 1000000 | 2 | 6327 | 3963 | 1.596518 |
| 1000 | 8 | 58 | 731 | 0.079343 |
| 10000 | 8 | 284 | 769 | 0.369311 |
| 100000 | 8 | 2500 | 790 | 3.164557 |
| 1000000 | 8 | 25183 | 4218 | 5.970365 |
| 1000 | 16 | 80 | 970 | 0.082474 |
| 10000 | 16 | 533 | 1054 | 0.505693 |
| 100000 | 16 | 5044 | 1828 | 2.7593 |
| 1000000 | 16 | 50813 | 8091 | 6.280188 |
| 1000 | 36 | 148 | 4445 | 0.033296 |
| 10000 | 36 | 1174 | 2290 | 0.512664 |
| 100000 | 36 | 11397 | 2376 | 4.796717 |
| 1000000 | 36 | 114036 | 8149 | 13.99386 |

We can notice that as the N value increases with the thread count increase the speed up ration is getting increased.

For the Smaller N Size the parallel execution doesn’t fit but for the larger N size the execution time is quite less since more than one core can perform the operation and the given task is divided among the multiple threads.

Though the parallel execution consumes some time for dividing the data and combining the results but the it is less time consuming if we work on large data sets.