| Table 1 | 2 | 5 | 10 | 20 | 100 |
|---|---|---|---|---|---|
| Speedup (Average over 5 trials) | 1.899752912 | 3.950298836 | 5.449732388 | 5.068478407 | 0.184948454 |

The increase in speedup peaks at around 10 threads. Based on the time difference table, the decrease in run time when compared to the previous number of threads continues until 10 threads. After 10 threads, we see an increase in time spent in the inner loop.

| Table 2 | 2 | 5 | 10 | 20 | 100 |
|---|---|---|---|---|---|
| Speedup (Average over 5 trials) | 1.934166099 | 4.463335748 | 8.190477216 | 13.64533739 | 1.72534533 |

The increase in speedup peaks at around 20 threads. Similar to the situation in table 1, based on the time difference table, the decrease in run time when compared to the previous number of threads continues until 20 threads. After 20 threads, we see an increase in time spent in the inner loop.

| inner loop: main computation time | setup: parallel overhead |
|---|---|

|  | t= | 2 | 5 | 10 | 20 | 100 |
|---|---|---|---|---|---|---|
|  | setup | 0.000247 | 0.000392 | 0.000534 | 0.001036 | 0.004907 |
| n=100 | inner loop | 0.000131 | 0.000144 | 0.000209 | 0.000353 | 0.008985 |
|  | setup | 0.000253 | 0.000346 | 0.000577 | 0.001486 | 0.003854 |
| n=1000 | inner loop | 0.000576 | 0.000433 | 0.001058 | 0.001553 | 0.036943 |
| (table 1) | setup | 0.000301 | 0.000339 | 0.000579 | 0.001083 | 0.004484 |
| n=10000 | inner loop | 0.024376 | 0.012397 | 0.007356 | 0.008341 | 0.246643 |
| (table 2) | setup | 0.000232 | 0.000319 | 0.00055 | 0.001266 | 0.003755 |
| n=100000 | inner loop | 1.467324 | 0.6061 | 0.359155 | 0.226608 | 1.869368 |

Code commented in the program is originally used to measure time. Average time per thread over 5 trials is measured for the first pragma statement that sets up the threads (only once). Similarly, average time per thread spent on individual nested loops over 5 trials is also collected, including the pragma omp for directive. Note that the extra clauses and directives required to setup the measurements add extra runtime. Also, the total time of the outer loop is not taken because it is mainly indicated by the time shown as the main part, and it scales up with problem size nonetheless.

Time difference compare to the previous number of threads

| 2 | 5 | 10 | 20 | 100 |
|---|---|---|---|---|
| 0 | 0.000145 | 0.000142 | 0.000502 | 0.003871 |
| 0 | 0.000013 | 0.000065 | 0.000144 | 0.008632 |
| 0 | 0.000093 | 0.000231 | 0.000909 | 0.002368 |
| 0 | -0.000143 | 0.000625 | 0.000495 | 0.03539 |
| 0 | 0.000038 | 0.00024 | 0.000504 | 0.003401 |
| 0 | -0.011979 | -0.005041 | 0.000985 | 0.238302 |
| 0 | 0.000087 | 0.000231 | 0.000716 | 0.002489 |

<table>
<tr><td>0</td><td>-0.861224</td><td>-0.246945</td><td>-0.132547</td><td>1.64276</td><td></td></tr>
<tr><td colspan="5">Setup time increases as the number of threads increases, while inner loop, where computations are performed, sees some decrease in run time until certain number of threads is reached. This is likely due to the saturation of threads, which do not get enough tasks. It seems that the differences in speedup ultimately depends more the inner loop before parallel overhead overtakes it.</td><td></td></tr>
</table>