**Second Assignment Report**

**Exercise 0**

**Comparison between touch first and touch by all approaches in the parallel sum of N numbers.**

First of all, the testing was done with N=2.000.000.000. Both approaches, as one would expect, are better in term of execution times when compared to the serial algorithm, which took approximately 12.4 seconds. Looking at the plot below, we see that the touch first approach takes two to four times more than the touch by all one.

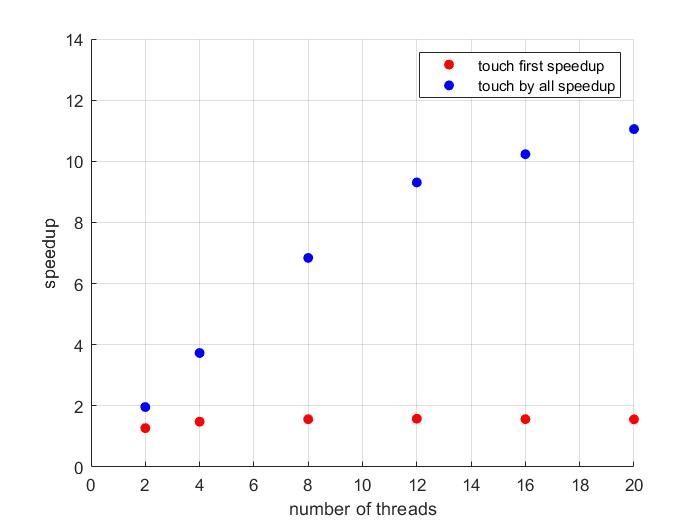
**Execution times for N=2.000.000.000**

Immagine che contiene cielo

Descrizione generata automaticamente

Therefore, we get the following plot for the speedup of the two codes:

**Strong scaling for N=2.000.000.000**

****

Notice that the touch first speedup basically remains the same while increasing the number of threads, and actually starts to decrease when having a larger number of threads due to parallel overhead. The touch by all speedup resembles more the behaviour predicted by Amdahl’s law, and in particular for 8 threads or less the speedup is almost linear, while using more threads as before increase the overhead. To estimate such overhead, first of all I ran the serial version of the codes and the parallel one with only one thread. For both versions the overhead associated with OpenMP is between 0.2 and 0.4 seconds

**Touch first serial estimation**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Threads | 2 | 4 | 8 | 12 | 16 | 20 |
| Speedup | 1.27 | 1.48 | 1.56 | 1.58 | 1.56 | 1.56 |
| Serial Time | 0.57 s | 0.57 s | 0.59 s | 0.60 s | 0.61 s | 0.62 s |

**Touch by all serial estimation**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Threads | 2 | 4 | 8 | 12 | 16 | 20 |
| Speedup | 1.96 | 3.73 | 6.84 | 9.31 | 10.23 | 11.05 |
| Serial Time | 0.019 s | 0.024 s | 0.024 s | 0.026 s | 0.038 s | 0.043 s |

**Strong Scaling for Nsearch=20.000.000, Ndata=100.000.000**

Immagine che contiene cielo

Descrizione generata automaticamente

