High Performance Computing Homework 8

Alejandra Torres Manotas

April 30, 2019

1 Explanation of files

1. In the point (b) we had to modify the code such that it stops when there are no more robots on the table, and it takes a command line argument which is the random seed. For this, it is necessary to run the file *serial.cpp* with the instruction wrote at the end of the file.

   And for the part (c) we have to write a script to be able to run four different initial conditions at the same time using GNU Parallel. For this, it is necessary to run first the script *script.sh*

2. In part (d), I speeded up the code using OpenMP, using default(none) in the file called *serialOMP.cpp*. And in the part (e) I wrote a job script, called *scriptOMP.sh*, to perform a scaling analysis, that runs the code for values of $OMP\_NUM\_THREADS$ ranging from 1 to 4.

3. Finally, in the part (f) I made a plot of the scaling (See Figure 1) and estimate the serial fraction (See Table 1)

$$f = \frac{T_s}{T_s + NT_1},$$

   where T_s denotes the time spent in the serial part of the computation, N is the number of threads and T_1 is time each thread takes to run. Thus $NT_1 = T_P$ denotes the time spent in the parallelizable part of the computation.

   In the picture I showed the threat time diminution when the number of threats is increasing.

Threats	Parallet Time	Threat Time	Serial Faction
1	0m7.461s	1.86525	0.7467757263100734
2	0m6.151s	2.05033	0.7815230517866023
3	0m5.362s	2.681	0.8040562762653023
4	0m8.334s	8.334	0.725285954445067

Table 1: Time measure and Serial Fraction calculation.

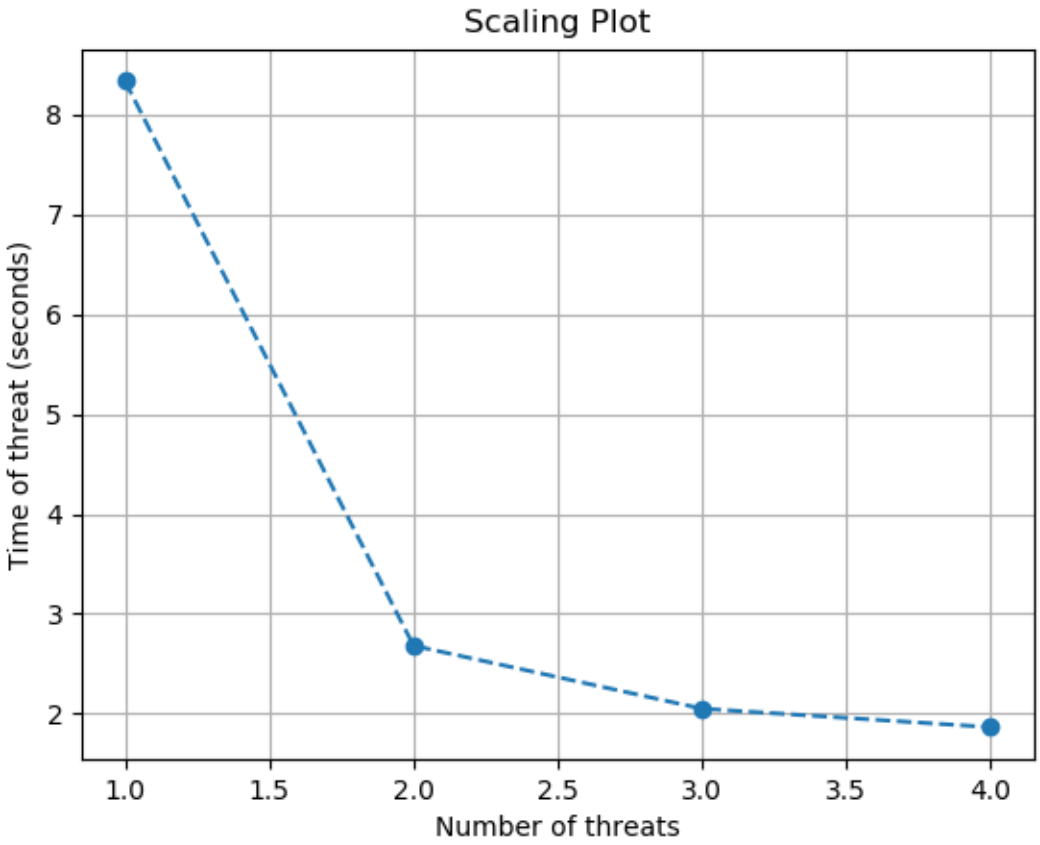

Figure 1: Scalling Plot of *serialOMP.cpp*