Giulia Franco
matricola SM3500370
Year 2018/2019
Exercise 6, High Performance Computing course.

Benchmark using MKL multithread library.

A benchmark is the act of running a computer program in order to assess the relative performance of an object. The exercise consist in compile the HPL benchmark against the mkl libraries and tune it in order to get close to theoretical peak performance of a node of Ulysses cluster.
HPL benchmark solves a random system of linear equations and reports time and floating-point execution rate using the following formula:

$$N_{FLOPS} = \frac{2n^3}{3} + 2n^2 \tag{1}$$

where n is the problem size.
The first part is downloading and installing HPL: this is done by following the instruction on the website. Once all the programs are set up, the aim is to find the right combination of parameters N (size of the problem), Nb (block size), P and Q (grid of MPI processes, where P represent the number of process rows and Q the number of process columns) in order to achieve the best peak performance. First let's vary the block size: Nb will take values (192,256,400,750,1000) while $N = 64512$, $P = 4$, $Q = 5$.

Results for HPL benchmark

Nb	GFlops(theoretical)	GFlops	Peak Performance (%)
192	**448**	**4.025e+02**	**89,75**
256	**448**	**3.970e+02**	**88,61**
400	**448**	**4.074e+02**	**90,84**
750	**448**	**3.868e+02**	**86,33**
1000	**448**	**3.526e+02**	**78,70**

It's possible to see that the peak performance is in each of the previous case, up to 99%.
We proceed trying the parameters on the Intel optimized benchmark, setting the problem size $N = 64512$, number of test $n = 2$, one trial to run and data alignment value (in Kbytes) to 4. The following table shows the obtained results.

Comparing results HPL-Intel benchmark

Nb	GFlops (HPL)	GFlops (Intel-1)	GFlops (Intel-2)
192	**4.027e+02**	**414.0183**	**413.9716**
256	**3.972e+02**	**415.8126**	**415.7015**
400	**4.076e+02**	**434.9973**	**435.3549**
750	**3.871e+02**	**433.9637**	**439.8435**
1000	**3.529e+02**	**415.9759**	**416.8664**

It's clear that the Intel benchmark returns better GFlops values than the HPL. This happens probably because the formal assign threads to processor better than mkl, in order to minimize communications and achieve a better peak performance.

Finally we can vary the number of threads and processor of the problem, using HPL benchmark (using *OMP_NUM_THREAD* command). The size of the problem is kept to $N = 64512$, $Nb = 192$, while P and Q are different as the number of threads changes.

HPL benchmark values (different number of threads and processors)

Threads	N. processors	GFlops (computed)	P	Q
1	**20**	**3.708e+02**	**4**	**5**
1	**20**	**4.264e+02**	**5**	**4**
2	**10**	**2.278e+02**	**2**	**5**
2	**10**	**2.267e+02**	**5**	**2**
4	**5**	**1.230e+02**	**1**	**5**
4	**5**	**1.231e+02**	**5**	**1**
5	**4**	**1.013e+02**	**1**	**4**
5	**4**	**1.015e+02**	**4**	**1**
5	**4**	**1.005e+02**	**2**	**2**
10	**2**	**5.020e+01**	**2**	**1**
10	**2**	**5.038e+01**	**1**	**2**
20	**1**	**5.038e+01**	**1**	**1**

From the previous results we can conclude that the peak performance (GFlops) decreases with the number of processors (as the number of threads increases). This is unespected: the performance should increase with the number of threads since the problem is better divided. So we can conclude that the program doesn't recognize the command, problem that we can might overcome using *MKL_NUM_THREAD* command.