Serie 3

Exercise 1 (Theoretical roofline)**.**
Allocate a node and run

```
$> srun -n 1 cat /proc/cpuinfo
```

To determine the theoretical peak performance of a core:

1. compute the theoretical performance of the memory
2. compute the roofline model, in particular the ridge point

Exercise 2 (measured roofline)**.**

- compile and run the code in Stream to compute the sustained memory performance
- compile and run the code in Dgemm to compute the sustained peak performance
- compute the roofline model, in particular the ridge point

How to run:

```
$> module load intel
$> export KMP_AFFINITY=compact,granularity=fine
$> export OMP_NUM_THREADS=7
$> srun -n 1 -N 1 -p serial --reservation=phpc2021 -A phpc2021 --cpus-per-task $OMP_NUM_THREADS ./stream
$> srun -n 1 -N 1 -p serial --reservation=phpc2021 -A phpc2021 --cpus-per-task $OMP_NUM_THREADS ./dgemm
```

How do you justify the difference?

Exercise 3 (Jacobi stencil)**.**
We want to solve the jacobi stencil:

```
u(i,j) = 1/4*(u(i-1, j) + u(i + 1, j) + u(i, j - 1) + u(i, j + 1)
```

1. What is the arithmetic intensity of this equation
2. According to the roofline model, what is the maximum performance we can get

Go to the directory Jacobi:

- `jacobi.c` is the main driver
- `jacobi-naive.c` is the "classic" implementation

Compile and run this code using the makefile.

1. What is the difference between the reported and the target performance?
2. How can performance be improved?

Exercise 4 (If you finish early...)**.**
`jacobi-sse.c` and `jacobi-avx*.c` contain different implementation using DLP with intrinsics.

1. Which one is the fastest and why? Can you beat those implementations ?