

## Complexité: Espace

The goal is to share the same vector between all processes, so the choice of algorithm won't change the results.

Let  $n$  be the number of processes and  $x$  the space taken by the vector in the system's memory:

$$O(n \cdot x)$$

## Complexité: Temps

Here we'll witness different complexities for each algorithms.

Simple: Here we have a simple code that uses the command *for* that iterates the same code for all processes, we have then  $(n-1)$  cycles. Let each cycle be of time  $t$ , and  $n$  processes:

$$O((n-1) \cdot t), \text{ or } \rightarrow O(n \cdot t)$$

Anneau: It's here we see the interest of parallélisme, we are sending the same amount of information, but since we do it in parallel we do less cycles. Let time per cycle be  $t$ , and  $n$  the number of processes:

$$O(2 \cdot t + (n-3)/2 \cdot t), \text{ or } \rightarrow O((n/2) \cdot t)$$

Hypercube: For this last one, the train of thought is the same, just that we have even more processes sending information:

cycle 1 -> we have 1 process  
 cycle 2 -> we have 1 process  
 cycle 3 -> we have 1 process  
 ...  
 cycle  $n$  -> we have  $2^{(n-1)}$  processes

For  $n$  processes we will require  $\log_2(n)$  cycles. time  $t$  per cycle

$$O(\log_2(n) \cdot t)$$

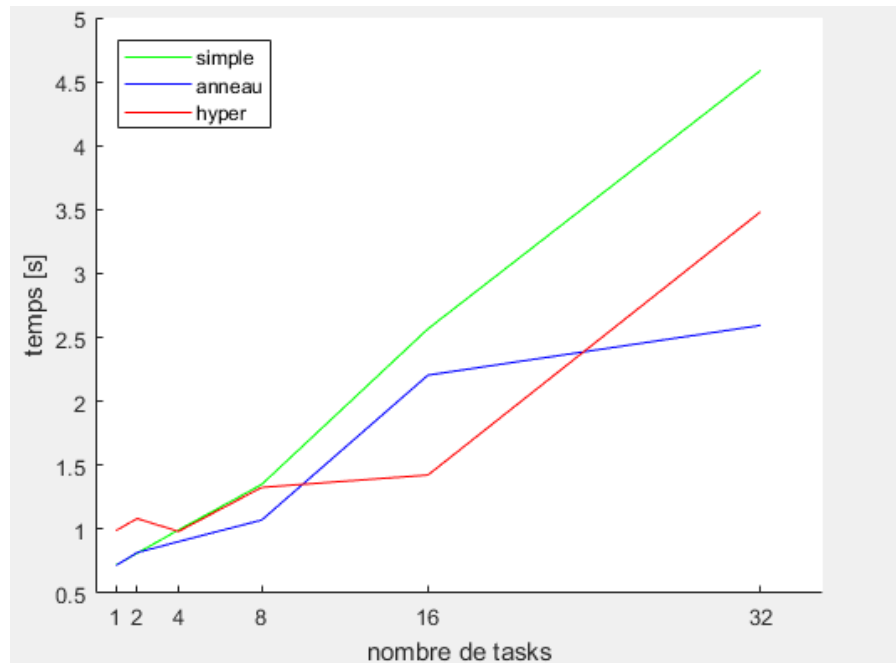
## Résultats

For the results I used the following code in the batchfile:

```
1  #!/bin/sh
2
3  #SBATCH --cpus-per-task=1
4  #SBATCH --job-name=broadCast
5  #SBATCH --ntasks= (insérer nombre de tasks -> [1,2,4,8,16,32,64])
6  #SBATCH --time=0-00:01:00
7  #SBATCH --mail-user=Joao.Costa@etu.unige.ch
8  #SBATCH --partition=debug-EL7
9  #SBATCH --output=output.out
10 #SBATCH --constraint=E5-2660V0
11
12 srun ./executable
```

I had problems to execute this batchfile with  $n_{\text{Proc}} = 64$ , I discussed them with you via email.

For simple, Anneau and Hypercube, I find unexpected results, as Anneau became faster than Hyper as  $n$  grew larger. Simple is as expected.



I don't know why but using the simple BCast command didn't work as intended, it was very slow, I don't know if it was just by chance, but I did try it many times, and I always got the same results

