

Nom: JODAR SOARES

Prénom: Gustavo

Print du read.me:

```
- Quel est le nombre de coeurs physiques de votre machine ?  
  Ma machine a 4 coeurs  
- Quel est le nombre de coeurs logiques de votre machine ?  
  8 coeurs logiques (2 threads par coeur)  
- Quelle est la quantité de mémoire cache L2 et L3 de votre machine ?  
  L2 - > 1 MiB  
  L3 - > 6 MiB
```

Mon premier approche était d'ajouter les cellules dans un array et d'utiliser MPI_Scatter pour diviser également entre les processus. Par contre j'ai reçu une erreur qui disait ne pas être possible car num_config n'était pas disponible pour afficher l'image. Alors j'ai décidé pour un approche plus simples: simplement diviser le for entre le nombre des processus:

```
cas_per_proc = int(nombre_cas/size)      You, now • Uncommitted changes  
for num_config in range(rank*cas_per_proc, (rank+1)*cas_per_proc):  
    t1 = time.time()  
    cells = np.zeros((nb_iterations, nb_cellules+2), dtype=np.int16)  
    cells[0, (nb_cellules+2)//2] = 1  
    for iter in range(1, nb_iterations):  
        vals = np.left_shift(1, 4*cells[iter-1, 0:-2]  
                               + 2*cells[iter-1, 1:-1]  
                               + cells[iter-1, 2:])  
        cells[iter, 1:-1] = np.logical_and(np.bitwise_and(vals, num_config), 1)  
    t2 = time.time()  
    compute_time += t2 - t1  
  
    t1 = time.time()  
    save_as_md(cells)  
    # save_as_png(cells)  
    t2 = time.time()  
    display_time += t2 - t1
```

Output:

1 processeur:

aff: 8.83767

cal: 0.720381

2 processeurs:

aff: 4.15411 , 4.56

cal: 0.347146 0.349923

3 processeurs:

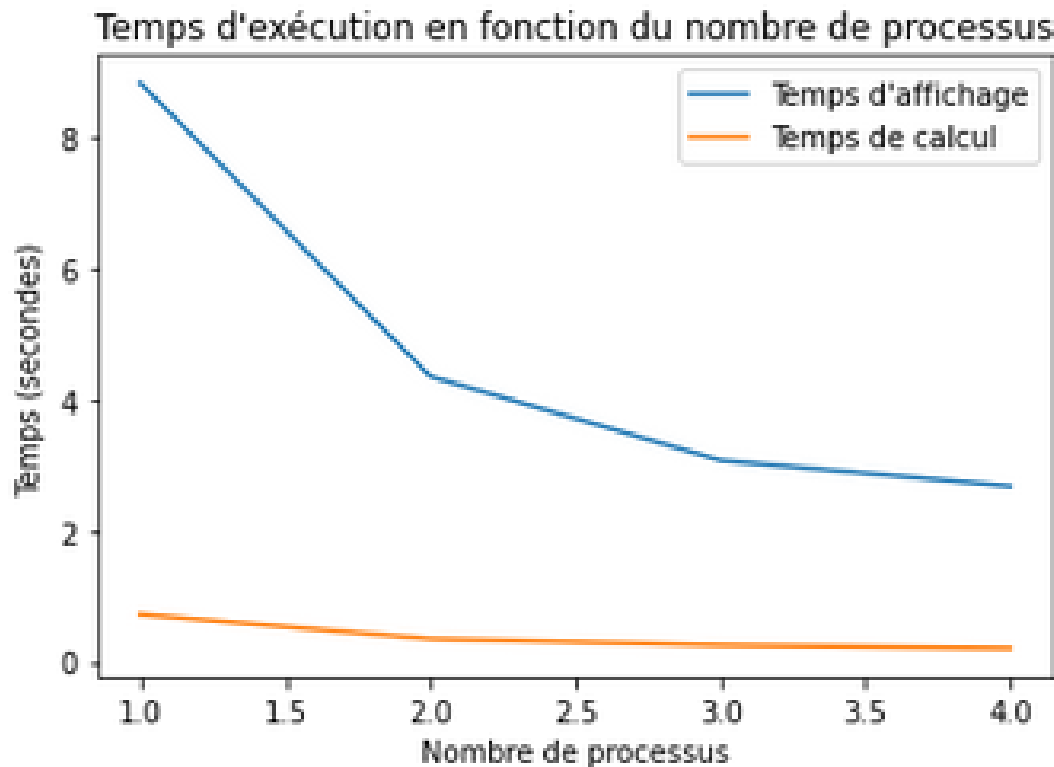
aff: 2.83444 3.0206 3.3709

cal: 0.238317 0.257159 0.260307

4 processeurs: aff: 2.47071 2.61014 2.63408 3.01499

cal: 0.190494 0.205707 0.203117 0.217412

temps d'affichage = $[8.83767, (4.15411+4.56)/2, (2.83444+3.0206+3.3709)/3, (2.47071+2.61014+2.63408+3.01499)/4]$, temps de calcul = $[0.720381, (0.347146+0.349923)/2, (0.238317+0.257159+0.260307)/3, (0.190494+0.205707+0.203117+0.217412)/4]$ o x c'est nombre de processus et y secondes



Calcul d'une enveloppe convexe

Pour la partie 1:

Code: `enveloppe_convexe.py`

Pour la partie 2:

Code: `enveloppe_convexe_2.py`