

# ST7 Projet Intel

Colonies de fourmis

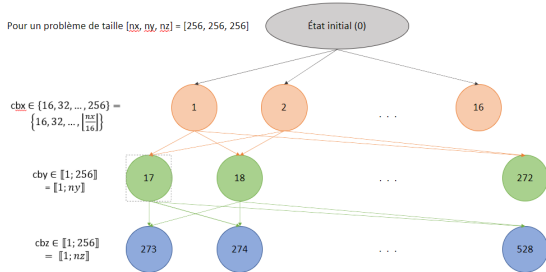
Simon Maréchal, Matthieu Oberon, Émile Prost  
Carlos Santos García, Paul Saurou

17 mars 2021

1. Modélisation du graphe
2. Stratégies implémentées
3. Parallelisation
4. Nos interrogations

# Modélisation du problème

Pour un problème de taille  $[nx, ny, nz] = [256, 256, 256]$



Matrice de phéromones  $\tau$  :  
 $\tau_{ij}$  la quantité de  
 phéromones sur l'arrête  
 allant du nœud  $i$  vers  $j$

$$\tau = \begin{matrix} & \begin{matrix} 1 & \underbrace{nx = 16} & \underbrace{ny = 256} & \underbrace{nz = 256} \end{matrix} \\ \begin{matrix} 1 \\ \vdots \\ nx = 16 \\ \vdots \\ ny = 256 \end{matrix} & \left\{ \begin{array}{cccc} & X & \dots & X \\ & & & X & \dots & X \\ & & & \vdots & \ddots & \vdots \\ & & & X & \dots & X \\ & & & & & X & \dots & X \\ & & & & & \vdots & \ddots & \vdots \\ & & & & & X & \dots & X \end{array} \right\} \end{matrix}$$

$$\tau \in \mathbb{R}^{273 \times 529}$$

# Stratégies implémentées

Voici les stratégies implémentées :

- Classic Ant System : Toutes les fourmis laissent des phéromones dans leur parcours du graphe
- Élitiste : Uniquement la meilleure fourmi laisse des phéromones dans le graphe
- Max-min Ant System : Idem qu'élitiste, mais les phéromones dans le graphe restent bornées

## À faire

Reste à implémenter Ranked Ant System où uniquement les k meilleures fourmis laissent des phéromones sur le graphe

# Code

```
comm = MPI.COMM_WORLD
NbP = comm.Get_size()
Me = comm.Get_rank()

tau = tau0 # initialisation de la matrice de pheromones
for iter in range(n_iter):
    paths = []
    costs = []
    # calcul de chemins et leurs couts pour une sous famille de fourmis
    for ant in range(nb_ants/NbP):
        path = compute_path(tau, ...)
        paths.append(path)
        costs.append(compute_cost(tau, ...))

    # gather des chemins et couts des sous familles de fourmis
    comm.Gather(paths, all_paths, root=0)
    comm.Gather(costs, all_costs, root=0)

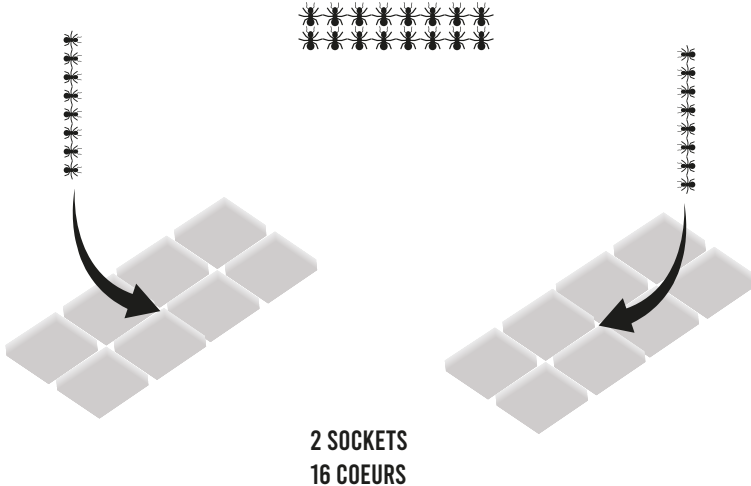
    # calcul de tau et du meilleur chemin et son cout
    if Me == 0 :
        best_path = ...
        best_cost = ...
        tau = compute_tau(tau, all_paths, all_costs, ...)
    # broadcast de tau
    comm.Bcast(tau, root=0)
```

Matrice de phéromones  $\tau$  initialisée sur tous les processus. Tant que le nombre maximal d'itérations n'est pas atteint :

- Chaque processus calcule les **paths** et **costs** des fourmis associées
- Gather : récupère les costs sur le processus 0, et met à jour  $\tau$
- Broadcast du nouveau  $\tau$  sur tous les processus

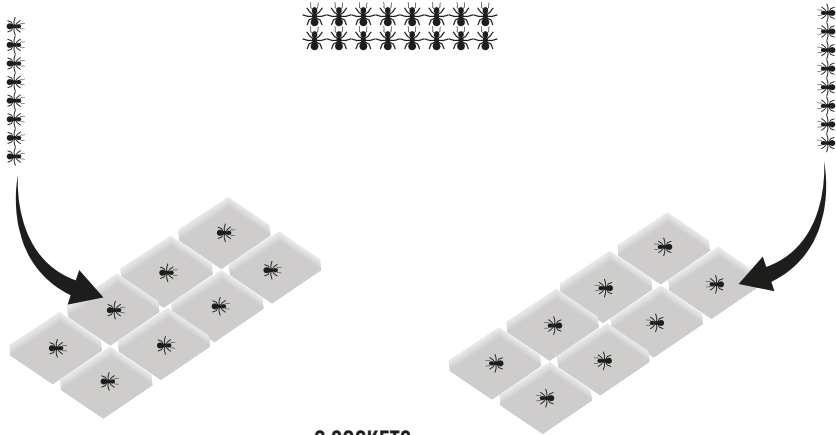
# Première version : Ants bind to socket

## SYSTEME DE COLONIE DE FOURMIS



# Deuxième version : Ants bind to core

## SYSTEME DE COLONIE DE FOURMIS



2 SOCKETS  
16 COEURS



# Nos interrogations

- Comment fonctionne sbatch ? (On a tenté mais les batch étaient coincés en Pending)
- Comment fonctionne iso3dfd ? (Comment sont répartis les threads ? Doit-on l'appeler avec mpi ?)
- Les paramètres de cache blocking ne semblent pas avoir d'effet sur les performances... sont-ils réellement des paramètres à optimiser ?

FIN