

Projet Fourmis

Thomas Varin

March 2024

Lorsque l'on ne parallélise que les fourmis, il est possible de le faire de manière à ce que chaque processus de rang non nul possède sa propre colonie. Dans mon cas, pour des raisons de simplicité du code, je n'ai traité que le cas où la division du nombre de fourmis tombe juste. Cependant il n'est pas très difficile d'adapter le programme, surtout qu'en python les fonctions MPI sont très permissives sur les dimensions des arguments. Alors, chaque processus possède sa propre colonie, qui doit donc obéir à la "physique" du programme, en ne se basant que sur les phéromones. Celles-ci sont mises en commun après chaque appel à la fonction "advance", ainsi qu'avant chaque affichage. Cette mise en commun se fait avec un allreduce, avec la fonction max. Il est facile de se convaincre que, hormis lors de la fonction "do_evaporation", les phéromones actualisées sont toujours augmentées, ce qui fait que la fonction max est appropriée.

Pour ce qui est du speedup, il dépend bien sûr de la taille du labyrinthe. Voici un tableau récapitulatif des valeurs mesurées.

Taille	Séquentiel	2 coeurs	4 coeurs	8 coeurs
25x25	0.27s	0.40s	0.32s	0.40s
50x50	0.90s	1.01s	0.65s	0.75s
100x100	3.41s	3.15s	2.34s	2.60s
200x200	13.3s	11.2s	7.61s	8.98s

Pour réaliser ces mesures, j'ai fait tourner mon programme sur 50 itérations, ce qui me semble suffisant pour faire négliger les bruits de mesure. Je remarque aussi que la plupart du temps, c'est le processus 0 qui finit le dernier, ce qui résulte probablement de la fermeture de pygame. De plus, avec 8 coeurs, le programme est plus lent qu'avec 4, probablement dû aux communications qui ne sont pas très efficace pour des données qui se trouve être assez petites.

Il est possible de plus paralléliser le code, notamment en gérer l'évaporation en parallèle, un peu à la manière du TP sur le jeu de la vie. J'ai aussi eu l'idée d'adopter une approche maître-esclave en partitionnant les lignes pour que chaque processus possède le même nombre de fourmis. Cette approche me semblait cependant trop compliquée à mettre en place, avec des résultats qui ne semblant être extraordinaires.