École polytechnique de Louvain

Rapport du Projet 1
LINFO1252

Projet 1 : Multithread et Spinlock

Auteurs :
Guillaume Jadin (NOMA : 10581800),
Nicolas Jeanmenne (NOMA : 487141900)

7 décembre 2021

Table des matières

1 Introduction **1**

2 Résultats des performances **1**

2.1 Problème des philosophes 1

2.2 Problème des producteurs-consommateurs 2

2.3 Problème des écrivains et lecteurs 3

2.4 Verrous par attente active 4

3 Conclusion **5**

1 Introduction

Ce rapport vise à présenter les performances observées de 3 programmes différents :
— Le problème des philosophes,
— Le problème des producteurs-consommateurs,
— Le problème des écrivains et lecteurs,
avec l'utilisation de 2 types de verrou (pour gérer la concurrence) :
— Les verrous POSIX,
— Les verrous actifs (QUICKSPIN).

2 Résultats des performances

2.1 Problème des philosophes

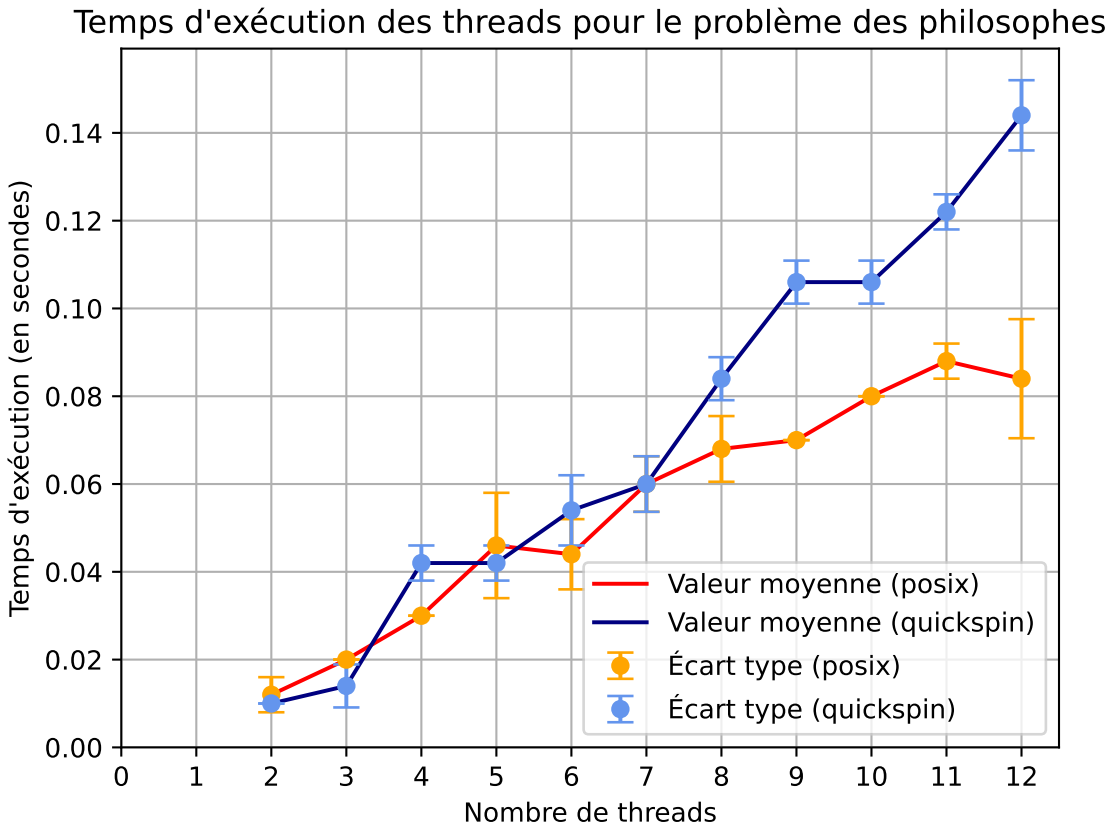

FIGURE 1 – Graphe des philosophes (POSIX & QUICKSPIN)

Nous pouvons remarquer que pour l'exécution du problème des philosophes avec 20 threads qui font 100 000 cycles penser/manger (1 philosophe = 1 thread), l'allure des courbes est globalement similaire. On notera toutefois que les mutex POSIX sont mieux optimisés lorsque l'on dépasse le nombre de threads pouvant s'exécuter simultanément sur nos systèmes.

2.2 Problème des producteurs-consommateurs

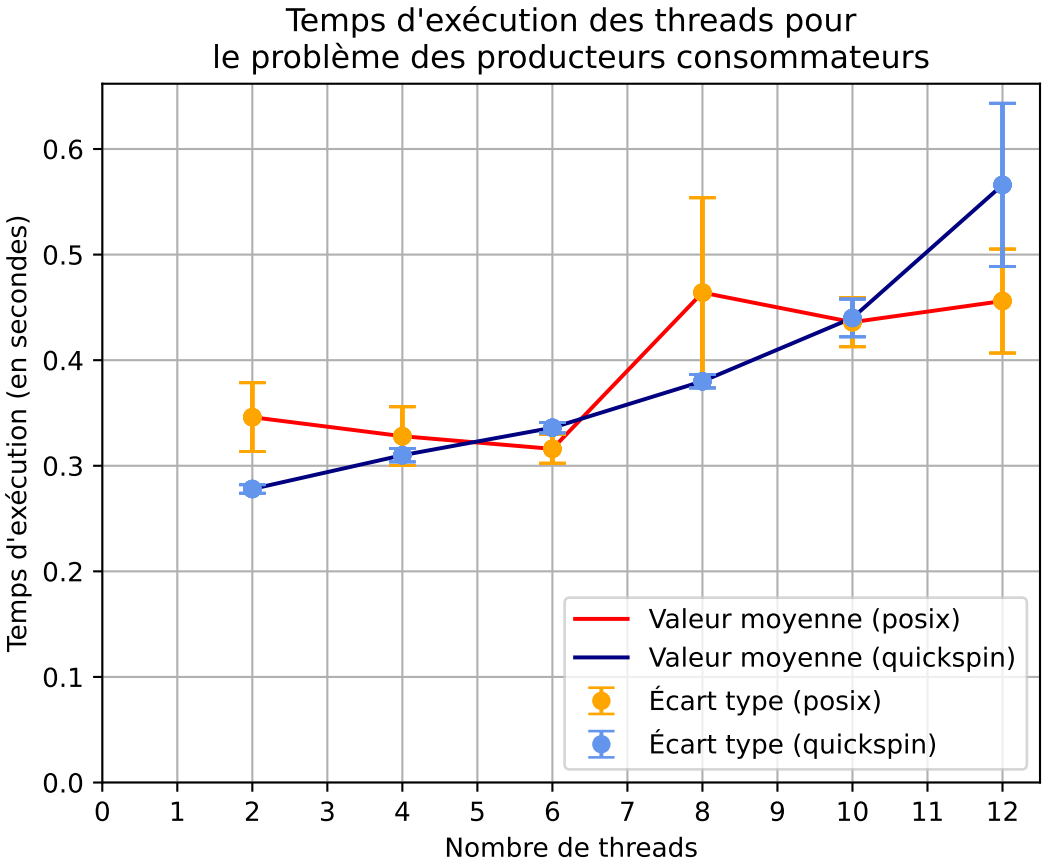

FIGURE 2 – Graphe des producteurs-consommateurs (POSIX & QUICKSPIN)

Pour le problème des producteurs-consommateurs, nous remarquons que les temps entre les verrous POSIX et QUICKSPIN sont relativement similaires.
De plus, nous constatons que le QUICKSPIN a une courbe croissante tandis que le POSIX possède certaines optimisations permettant d'améliorer (ou de détériorer) la vitesse du producteur-consommateur en fonction du un nombre de threads.

2.3 Problème des écrivains et lecteurs

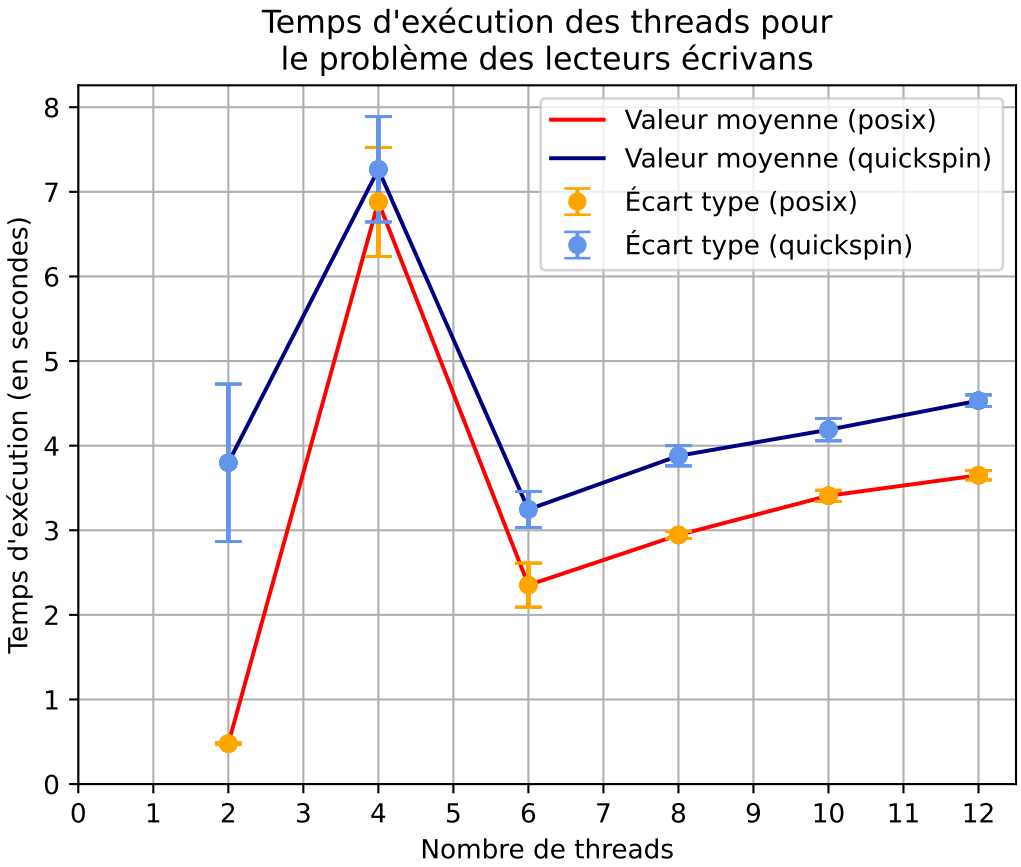

FIGURE 3 – Graphe des écrivains et lecteurs (POSIX & QUICKSPIN)

En ce qui concerne le problème des écrivains et lecteurs, le graphe nous démontre que les verrous POSIX et QUICKSPIN évoluent de la même manière même si POSIX a un temps d'exécution plus court que QUICKSPIN.

2.4 Verrous par attente active

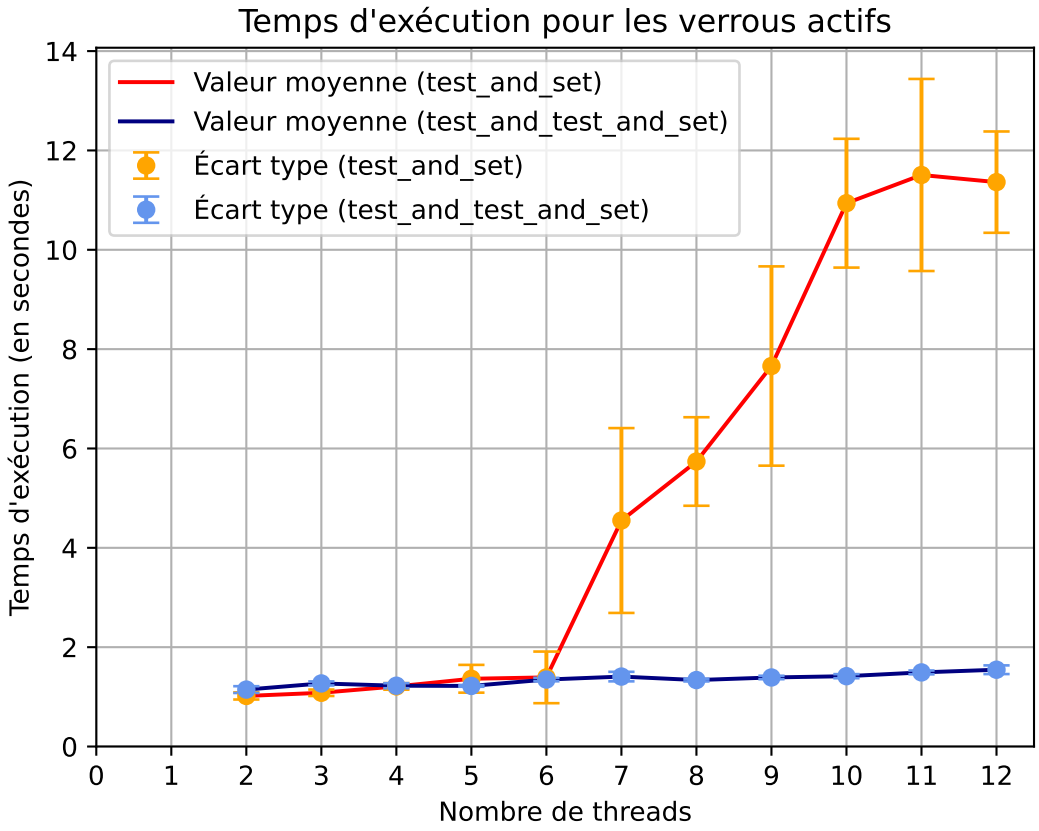

FIGURE 4 – Graphe des écrivains et lecteurs (POSIX & QUICKSPIN)

Lors de la comparaison des verrous actifs implémentant les algorithmes test_and_set (ts) et test_and_test_and_set (tts), nous pouvons constater que l'algorithme ts est beaucoup plus efficace que l'algorithme tts.
Cela est due au fait que l'algorithme tts utilise les bienfait de la mémoire cache en allant récupérer la valeur du lock au lieu de faire des swaps atomiques, opération très coûteuse pour le processeur.

3 Conclusion

Pour conclure ce projet, notre binôme a appris différentes méthodes pour protéger les zones sensibles stockées en mémoire. Nous avons vu que la librairie POSIX n'est pas la seule qui existe pour réaliser des mutual exclusion et des sémaphores.

Concernant les verrous par attente active, nous avons pu nous rendre compte que l'algorithme test_and_set consomme beaucoup plus de temps pour s'exécuter et qu'il n'est donc pas optimal par rapport à l'algorithme test_and_test_and_set.

En plus des compétences en C, on a aussi acquis des compétences dans l'écriture de scripts bash pour mesurer le temps d'exécution de nos programmes et transmettre ces données sur un graphe en utilisant Python. Ce projet aura été très enrichissant à réaliser et est une réelle plus-value dans le cadre de nos études en sciences informatiques.