Millian Poquet
Maître de conférences
Univ. Toulouse, IRIT, équipe Sepia
IRIT2-469, 120 cours Rose Dieng-Kuntz, 31400 Toulouse
millian.poquet@utoulouse.fr

Université de Toulouse

Structures de données pour ensemble de ressources en Rust

Contexte

Une plateforme de calcul à haute performance est un ensemble de machines très puissantes connectées par un très bon réseau. Les utilisateur·ice·s n'utilisent pas directement ces machines, mais passent par un logiciel appelé gestionnaire de ressources pour demander l'exécution de leurs tâches. Ces gestionnaires de ressources sont des codes distribués critiques qui prennent des décisions importantes, en particulier pour allouer *au mieux* les ressources aux utilisateur·ice·s, c'est-à-dire en essayant d'optimiser la satisfaction des utilisateur·ice·s, leur équité d'accès aux ressources, la consommation d'énergie… Ces gestionnaires de tâches ont en leur cœur un algorithme d'ordonnancement chargé de prendre beaucoup des décisions de gestion.

De nombreux algorithmes d'ordonnancement existent. La quasi totalité de ces algorithmes passent leur temps à manipuler quelles ressources étaient, sont ou seront disponibles pour prendre leurs décisions. Les opérations réalisées sur les ressources sont ensemblistes. Si une tâche j vient de finir d'utiliser les ressources A_j, et que B représente les ressources disponibles avant la fin de j, alors les ressources actuellement disponibles sont $A_j \cup B$. De manière similaire, un algorithme qui cherche des ressources libres sur le long terme peut partir des ressources actuellement disponibles et réaliser un ensemble d'intersections ensemblistes avec les ressources réservées dans le futur.

Objectifs du projet

Le but de ce projet est d'implémenter en Rust différentes structures de données permettant de gérer des ensembles de ressources, puis d'étudier et de comparer leur performance. Des structures variées comme des tables de hachage, des arbres binaires de recherche, des arbres équilibrés (arbres rouge-noir et AVL), des bitsets ou des ensembles d'intervalles seront étudiés. Toutes les implémentations devront suivre la même API (*trait* Rust). Il est attendu que vous définissiez cette API avec soin. Il est attendu que vous testiez que chacune de vos implémentations est fonctionnelle, en les soumettant toutes aux mêmes jeux de tests – des jeux de tests aux limites existent déjà pour le problème. Vous évaluerez la performance de vos implémentations en les exécutant sur des benchmarks faisant varier différents paramètres des ensembles de ressources comme le nombre total maximum de ressources ou leur fragmentation. La performance de dé/sérialisation des structures de données sera aussi évaluée.

Licences, droit d'auteur, propriété intellectuelle

Les implémentations réalisées seront faites sous licence libre ; en Apache-2.0 si possible ou dans une autre licence libre si le projet d'origine en impose déjà une. Les données issues de l'expérience, les documentations, rapports et articles seront écrits sous licence libre CC-BY. Un document de cession de droit d'auteur à l'UT sera signé pour pérenniser la liberté d'accès et de modification de vos productions.

Candidature

Pour maximiser vos chances de candidature, contactez-moi par mail avec les informations suivantes.

- Très courte motivation par rapport au sujet (2 phrases)
- Bulletins de notes (master et licence)
- CV court (parcours d'études, expériences professionnelles, compétences)

Si vous candidatez en groupe à un projet, merci de ne m'envoyer qu'un seul mail pour tout le groupe.