

Batch Scheduling in 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.

Batsim est un simulateur d'infrastructures de calcul centré sur l'étude de politiques de gestion de ressources. En pratique la simulation est la manière la plus utilisée pour expérimenter sur les politiques de gestion car il est nécessaire d'évaluer les politiques longtemps sur un grand nombre de ressources pour valider les algorithmes, ce qui aurait un coût pharaonique sur plateforme réelle en plus de la presque impossibilité de modifier un logiciel critique en production. Une interface Rust à Batsim a récemment été développée pour permettre d'implémenter les algorithmes de gestion de ressources en Rust.

Objectifs du projet

Le but de ce projet est de permettre d'implémenter de manière *conviviale* des politiques de gestion de ressources en Rust dans Batsim. L'interface permettant de parler à Batsim ayant déjà été réalisée, le cœur de votre contribution sera l'ensemble de structures de données et de fonctions permettant de prendre des décisions. Il est attendu que vous implémentiez les différents algorithmes de gestion de ressources *de base* utilisées sur les plateformes de calcul à haute performance : FCFS, EASY backfilling et Conservative Backfilling. Il est attendu que différentes structures de données dites *profils de backfilling* soient implémentées avec soin et mises sous tests d'intégration. La performance de ces structures de données devra être analysée et comparée à une implémentation existante en C++.

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.