UNIVERSITÉ TOULOUSE III PAUL SABATIER

Millian Poquet
Maître de conférences
Univ. Toulouse III, IRIT, équipe Sepia
IRIT2-469, 120 cours Rose Dieng-Kuntz, 31400 Toulouse
millian.poquet@univ-tlse3.fr

Rust: Are we SimGrid yet?

Contexte

Batsim est un simulateur d'infrastructures informatiques de calcul distribué spécialisé sur l'étude de politiques de gestion de ressources. Une partie importante des politiques des gestion de ressources est l'algorithme de *scheduling* qu'elles utilisent. Batsim est un programme en C++ construit au-dessus de la bibliothèque de simulation SimGrid, dont Batsim utilise les modèles de calcul et réseau. Batsim est généralement utilisé par son interface réseau (une socket ZeroMQ), ce qui permet d'y connecter des codes de prise de décision et de contrôle de la simulation, quel que soit le langage utilisé pour développer ces codes. Batsim a été initialement conçu afin d'étudier des politiques de gestion de centres de calcul à haute performance. Batsim fournit différents modèles afin qu'une utilisatrice puisse définir le comportement de ses applications.

L'architecture des dernières versions de SimGrid est similaire à celle d'un système d'exploitation pour CPU monocœur. Chaque acteur de la simulation peut être vu comme un processus utilisateur. Les processus ne s'exécutent jamais en même temps. Les processus peuvent agir avec le monde simulé par des *appels de simulation* (équivalent des appels système). Le *kernel* de SimGrid prend alors la main, réalise les tâches qu'il doit/veut faire, décide du prochain temps de la simulation puis réveille un des acteurs. Cette boucle de fonctionnement continue jusqu'à la fin de la simulation.

Objectifs du projet

Batsim est un projet de taille modeste, mais sa maintenance n'y est pas triviale de par la dette technique accumulée par certains choix technologiques. En particulier, l'utilisation de C++ a pu conduire a certaines situations de bugs mémoire *borderlines*. Réécrire Batsim en Rust est faisable et pas très coûteux, mais n'aurait qu'un intérêt très limité car le modèle d'exécution de Batsim est complètement contrôlé par SimGrid, qui réalise du scheduling des acteurs en *user space* en modifiant directement la pile d'exécution du processus de simulation — ce qui brise toute propriété mémoire intéressante que gagnerait Batsim en passant à Rust.

L'objectif principal de ce projet est de réaliser une veille technologique des bibliothèques de programmation asynchrones de Rust, afin de déterminer si une réécriture de SimGrid en Rust est faisable et à quel coût. Plus précisément, voici une liste indicative des tâches de ce projet :

- Prendre en main la version C++ de SimGrid pour comprendre son modèle d'exécution d'acteurs.
- Évaluer la performance des bibliothèques Rust de programmation asynchrone par acteurs, et vérifier que le modèle d'acteurs utilisé est comparable à celui utilisé par SimGrid. Comparer cette performance à celle de SimGrid sur des cas simples similaires.
- Implémenter une preuve de concept du cœur de fonctionnement de SimGrid en Rust : permettre l'exécution déterministe d'un ensemble d'acteurs, en exclusion mutuelle, en permettant aux acteurs de s'échanger des messages par un système d'*appels de simulation*. Les modèles complexes de simulation de SimGrid seront totalement ignorés dans cette implémentation : la preuve de concept utilisera des modèles de calcul et réseau simplistes.