

Les langages de programmation orientée objet Rust et C++: analyse et évaluation des performances

Sid TOUATI*

2020-2021

Lieu

Laboratoire INRIA Sophia Antipolis, France. Equipe-projet KAIROS.

Prérequis

Programmation orientée objet.

Présentation générale

Tout le monde connaît le langage de programmation C++, ses atouts, ses différentes versions (le C++ original, suivi de C++ 2011, suivi de C++ 2014), ses différents compilateurs (gnu C++ compiler, intel C++ compiler, etc.). En revanche, connaissez vous le langage de programmation orientée objet Rust ? Vous diriez "encore un autre langage qui ressemble à C++". C'est ce que je pensais, jusqu'à ce que des programmeurs avertis m'en fassent la démonstration que c'est un langage de qualité.

Comme C++, le langage Rust est un langage compilé; le code généré est un binaire exécutable sur un processeur physique, ce qui n'est pas le cas avec JAVA. En revanche, contrairement à C++, Rust offre des mécanismes de langage qui permettent de garantir que le code généré ne provoque pas d'erreur sur pointeur comme *segmentation fault*. Le langage de programmation garantit que chaque pointeur est valide. Aussi, le langage Rust offre des mécanismes de parallélisme de *threads* qui évite les *race condition*, ce qui n'est pas proposé en C++. D'après les testeurs de Rust, il semblerait que ce langage est très agréable à appréhender et à manipuler. Les concepteurs de Rust ont bénéficié d'une longue expérience avec les autres langages de

*Professeur à l'université Côte d'Azur. Sid.Touati@inria.fr

programmation comme C++, leur permettant de prendre le meilleur de ces langages, et enlever ce qui ne convient pas.

Comme tous les langages jeunes, il y a relativement peu d'applications qui sont écrites avec Rust. Mais comme tous les langages de programmation de qualité, il y a un début à tout. La première critique qui viendrait à l'esprit est la question des performances des applications écrites avec Rust, comparées à celles écrites avec C++. Les premiers programmeurs avec Rust affirment que les performances sont comparables, mais cela mériterait une étude plus sérieuse que des affirmations par l'exemple.

Objectifs du stage

Ce stage a pour objectif d'évaluer les performances de quelques applications C++ et Rust, les comparer afin de vérifier si les compilateurs Rust et C++ sont au même niveau d'optimisation. Il offre à l'étudiant une première expérience de spécialisation dans ce domaine de performances de codes, et de programmation avec le langage Rust. Le stage est constitué des tâches suivantes:

1. Étudier le langage Rust et faire une comparaison de ses concepts avec C++.
2. Programmer quelques algorithmes de graphes en C++ et en Rust.
3. Programmer quelques fonctions de calculs matriciels en C++ et en Rust.
4. Programmer quelques fonctions parallèles (en OpenMP) en C++ et en Rust.
5. Faire une étude des performances, en utilisant plusieurs options de compilation.

Lien pour commencer

- Apprendre Rust
<https://www.Rust-lang.org/fr/learn>