

## Démon de contrôle de processus distant en Rust

### Contexte

Il existe différentes manières d'étudier les systèmes et applications distribuées en informatique. Une de ces manières est l'expérimentation *réelle*, où le cœur d'une expérience consiste à exécuter de manière contrôlée de *vraies* applications distribuées en les lançant sur de *vrais* systèmes, tout en observant le comportement des applications et des systèmes pour en extraire des connaissances après analyse.

De nombreux outils existent pour lancer des programmes à distance. Surprenamment, peu de ces technologies sont satisfaisantes pour la réalisation d'expériences scientifiques contrôlées, que ce soit par leur manque de robustesse, par leur absence de remontée des erreurs, par leur manque de fonctionnalités ou à cause du bruit qu'elles induisent sur les mesures faites pendant les expériences.

### Objectifs du projet

Le but de ce projet est de développer une application distribuée légère et robuste pour exécuter et contrôler l'exécution de processus sur un ensemble de machines. L'architecture utilisée sera basée sur des *démons* : un processus à durée de vie longue sera exécuté sur chaque machine et fournira une interface de contrôle distant de la machine.

Les objectifs principaux de ce projet sont les suivants :

- Développer en Rust un démon pour contrôler les processus de la machine sur laquelle il s'exécute. Le code utilisera les mécanismes de programmation asynchrone de Rust, probablement via **tokio**.
- Concevoir un protocole réseau applicatif qui permette de contrôler le démon depuis une autre machine, et qui permette de faire remonter diverses informations relatives à l'exécution des processus et de l'état de la machine. Les données seront sérialisées par une technologie classique afin de ne pas obliger les processus de contrôle à être également en Rust. Les piles réseau utilisées seront probablement du TCP sur IP classique, mais d'autres piles plus efficaces pourront être utilisées si besoin.
- Mettre en place une démarche de qualité.
  - Écrire des tests de fonctionnement automatiques du démon.
  - Automatiser le lancement de ces tests via GitLab CI.