

Retours (biaisés) sur le langage Rust

François-Xavier Pineau¹

¹Centre de Données astronomiques de Strasbourg

23 mars 2021



□ Rappels sur Rust

- OV France 2019: “Le langage Rust: concepts et exemples”
- Rust combine:
 - contrôle de bas niveau: **performance**
 - fonctionnalités de haut niveau: **confort d'utilisation**
 - environnement **moderne**: Cargo (config toml), crates.io, doc markdown, tests
- **Performances de C/C++**, mais **sûr**
 - comme Java: **si ca compile, ca marche**, mais **sans GC**
 - compilé: temps de démarrage = lancement d'un process
- Le coût: contraintes du **RAII**
 - le compilateur ne laisse rien passer!
 - au début: on est obligé de se poser des questions, mais...
 - **avec un peu d'expérience**:
 - on apprend à mieux coder
 - **on se repose sur le compilato** (et on gagne du temps) au lieu de se battre contre

□ Quoi de neuf depuis?

- “Most loved language” sur Stack Overflow depuis 5 ans
- Plan de licenciement de [250 personnes](#) chez [Mozilla](#) en aout 2020, mais...
- Equipe dédiée chez Amazon pour [AWS](#)
- Equipe dédiée chez Microsoft, cf [ici](#) ou [ici](#)
- Transfert de propriétés de Mozilla à la récente [fondation Rust](#)
 - “Microsoft, Google, AWS, Huawei et Mozilla s’associent pour créer la Fondation Rust”
- Toujours en prod dans **Firefox**, **Dropbox**, [Facebook Libra](#) ...
 - rapide est sûr => populaire dans le monde la cryptomonaie
- Fait son entrée dans [le noyau linux](#) pour le dév. de [drivers](#)
- Fait son chemin pour entrer dans [curl](#)

□ Ecosystème

Petite sélection:

- [bindgen](#): “automatically generates Rust bindings to C libs”, ex [fitsio](#)
- [PyO3](#): “running and interacting with Python code from a Rust binary, as well as writing native Python modules”
- [Wasmbingen](#): “high-level interactions between Wasm modules and JavaScript”
- CLI: voir [clap](#) et/ou [structopts](#)
- Serveur Web: voir [arewewebyet](#): “no dominant framework at the level of Django or Rails”, “frameworks are smaller and modular, similar to Flask or Sinatra”
 - voir [actix-web](#), [rocket](#), ...
- GUI: voir [areweguiet](#) et [ce post](#). Bindings and wrappers for [GTK+](#), ...
- ...

Xmatch prototype

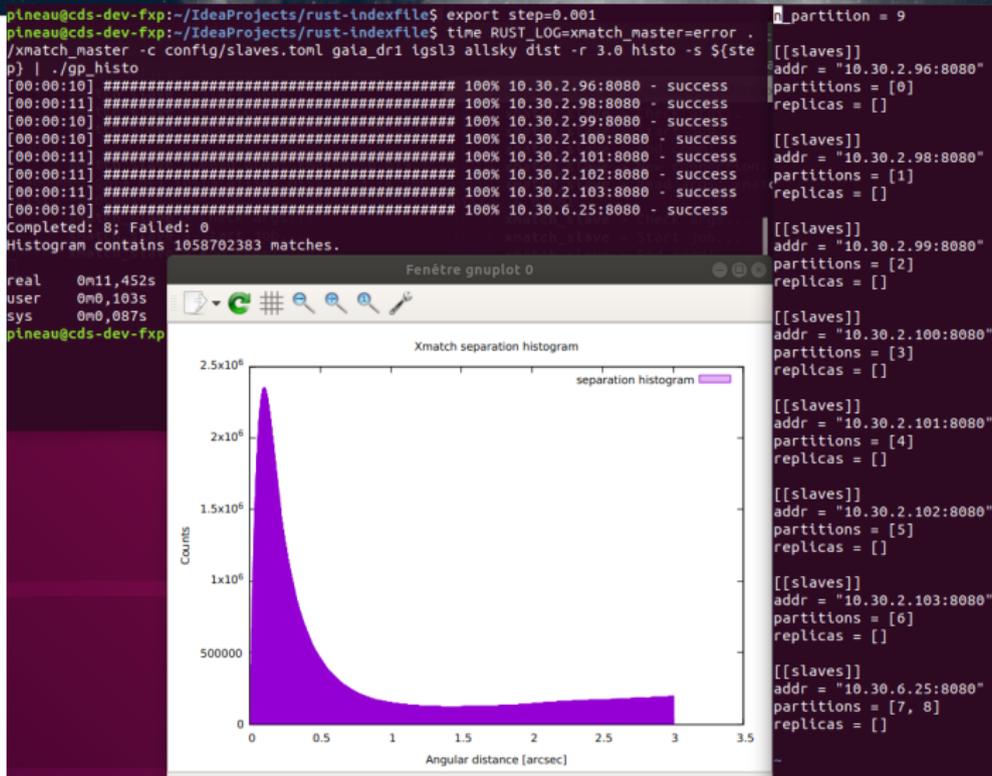


Figure 1: Xmatch

□ Le cas de CDS HEALPix

- Un même coeur en Rust, cf [github](#)
 - Prototype de cross-match et d'accée aux grands catalogues en Rust (interne)
 - Des librairies Python (cdshealpix Python, MOCPy)
 - Fonctions dans PSQL (pour SIMBAD et Vizier? En cours. . .)
 - pg-sphere?
 - Aladin Lite WebGL (en WebAssembly)

□ Freins à l'adoption

- Apprendre un nouveau langage avec une courbe d'apprentissage raide au début
 - être convaincu et motivé
 - avoir un peu de temps (il faut 1 mois pour être productif)
 - (ne pas se battre mais s'aider du compilateur!)
- Écosystème moins riche que Python (pas d'Astropy)
 - mais wrapper Python: PyO3 (Rust <-> Python)
 - binding Rust existent pour l'API C de NumPy, ...
- Communique difficilement avec Java (ok via JNI, comme C, C++)
- Niche de C/C++. Ok pour:
 - code critique avec besoin de performances (et/ou de sécurité)
 - CLI (Command Line Interface)
 - bibliothèques utilisables depuis plusieurs langages
- ...