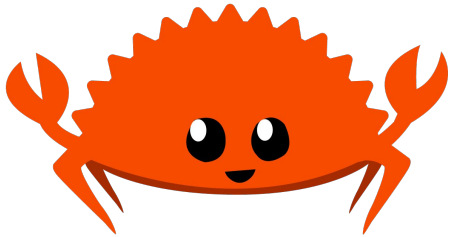
# GNU Parallel en Rust



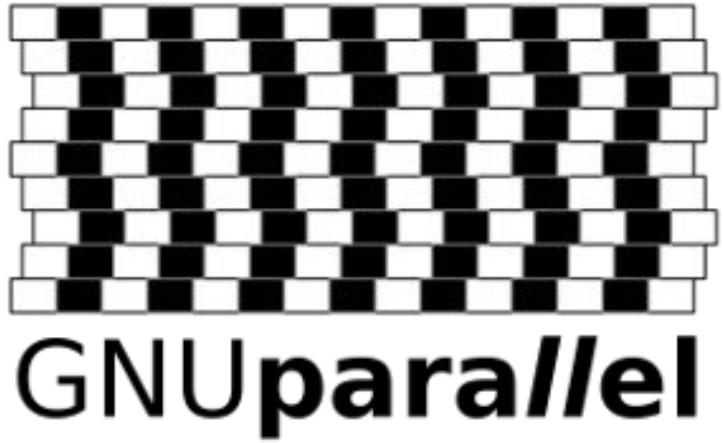
Soutenance de fin de projet du 06/04/2021

Groupe 14
Yaël Para - Victor Malod - Lucas Chaloyard - Dorian BARET

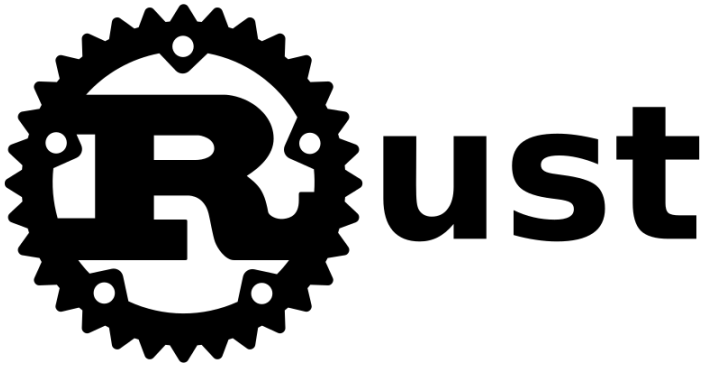
# Présentation générale

Rappel de l'objectif : écrire en Rust un outil similaire à GNU parallel.

- Comprendre le fonctionnement de GNU Parallel
- Développer et mettre en oeuvre un outil similaire avec le langage Rust



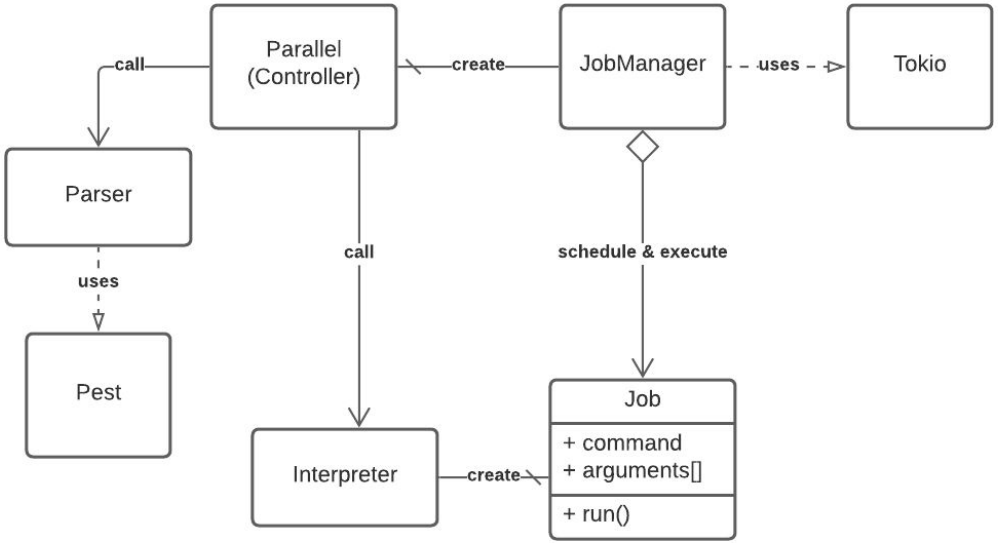

+



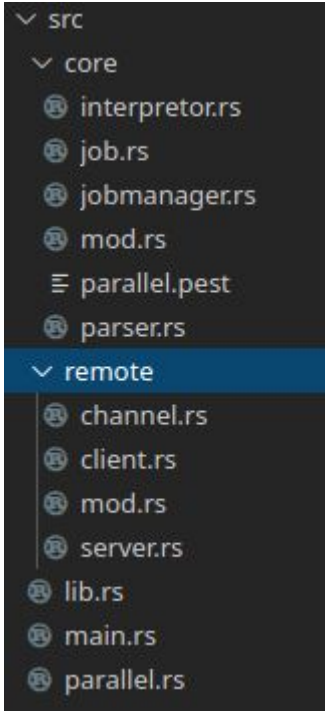
# Expression du besoin

- Implémentation du coeur de l’appli
- Implémentation de quelques options
  - dry run (affichage des commandes à exécuter)
  - keep order (ordre FIFO de la sortie des processus)
  - jobs (choix du nombre de thread à utiliser)
  - remote (exécution sur une machine distante)
- Interprétation de la grammaire de GNU Parallel
  - séparateurs
  - caractères spéciaux
  - format de commande

# **Solution** - Extrait de la conception



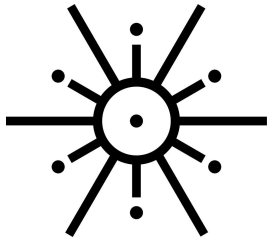

Architecture adoptée :



# **Solution** - Technique

Frameworks utilisés :

- Tokio : asynchronous framework



- Pest : parser framework

# **Solution** - Travail réalisé

- Parser
- Interpréteur
- Job Manager
- Client & Serveur
- Entrée du programme
- Veille technologique

# **Solution** - Performance

**Test simple de performance afin d'avoir une idée des capacités de notre outil.**

**Simulation de tentative de Brute-force d'un passcode à 6 chiffres (chaque chiffre compris entre 1 et 6)**

**Rust Parallel**

```
real    0m16.199s
user    1m25.678s
sys     0m54.129s
```

**GNU Parallel**

```
real    1m31.476s
user    1m50.453s
sys     1m33.664s
```

**Sequential** *(simple bash script)*

```
real    28m56.836s
user    10m51.210s
sys     18m34.264s
```

# Conclusion

- Rust Parallel : reverse engineering
- Rust Langage
  - Rapide / Efficace / Sécurisé
  - Difficile à prendre en main, mais il le vaut bien


Merci pour votre attention