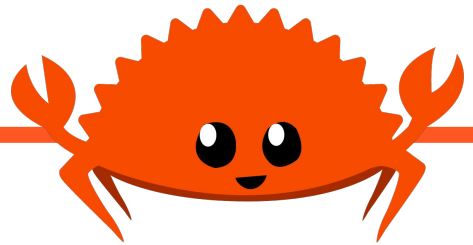
Rust Parallel



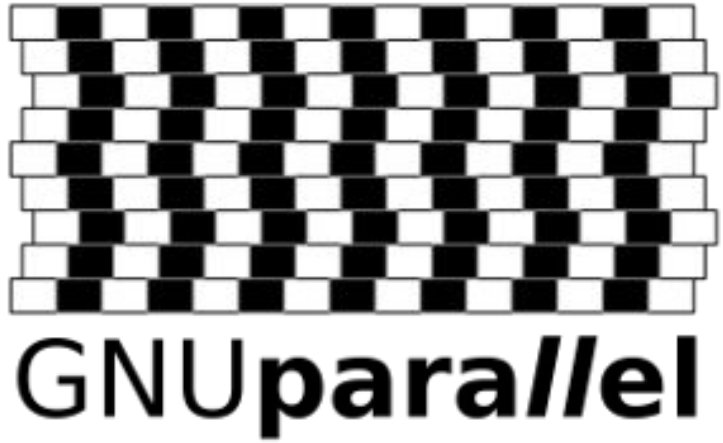
Soutenance de fin de projet du 06/04/2021

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

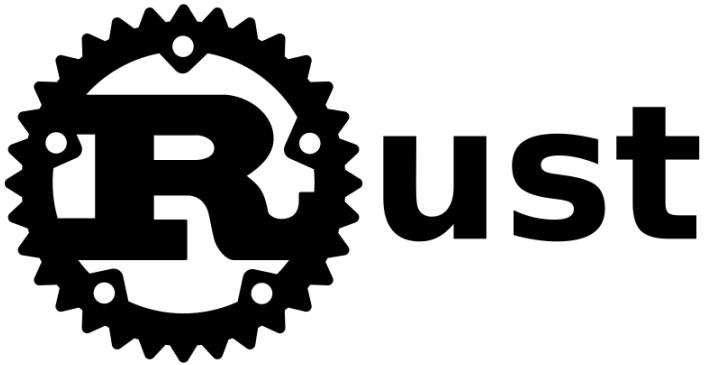
Présentation générale

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

Avoir une application qui reproduit le travail effectué par GNU Parallel. Pour cela nous avons besoin de réaliser plusieurs fonctionnalités.

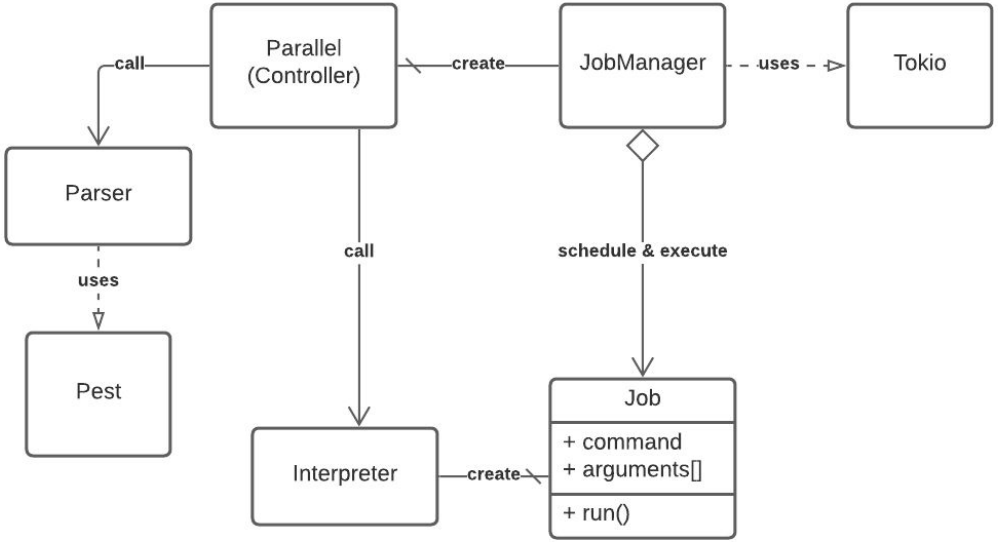
Principales fonctionnalités

- Création de "jobs" en fonction de la saisie utilisateur :
  - Parser la commande saisie par l'utilisateur.
  - Interprétation de cette dernière (prendra en compte les **options** et les **valeurs** des *séparateurs*).
- Implémentation de certaines options de base qui existe dans GNU Parallel (--dry_run, --keep_order, etc.)

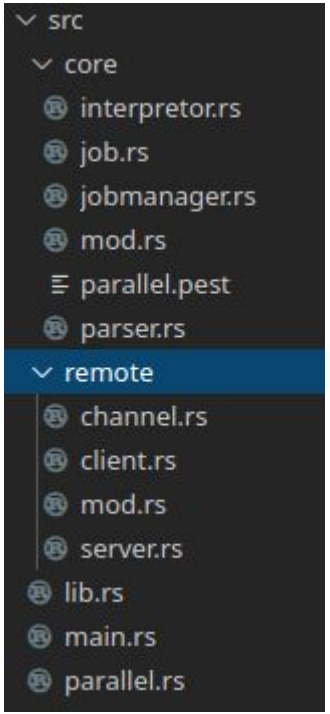
Fonctionnalités secondaires

- Lire l'entrée standard depuis un tube
- Exécution à distance
- Transfert de fichier dans le cas précédent

Solution - Extrait de la conception



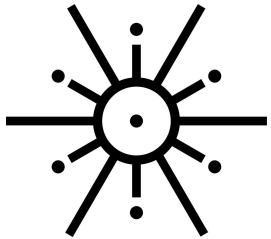
Architecture adoptée :



Solution - Technique

Frameworks utilisés :

- Tokio : asynchronous framework



- Pest : parser framework

Solution - Travail réalisé

- Veille technologique

- Parser
- Interpréteur
- Job Manager
- Client & Serveur

extrait de la grammaire :

```
main = {options* ~ commands* ~ separators*}

options = {
  ^"--dry-run"
| ^"--keep-order"
| ^"--pipe"
| ^"--jobs" ~ ASCII_DIGIT+
| ^"-j" ~ ASCII_DIGIT+
| ^"--server" ~ ASCII_DIGIT+
| ^"--client" ~ string ~ ASCII_DIGIT+
| ^"--help"
}
```

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