

I) Introduction

Cet article a pour sujet le langage Rust et ses utilisations dans les projets de grosses entreprises. Ce sujet me semble important car Rust devient de plus en plus populaire auprès des développeurs pour de nombreuses raisons, au point où il arrive depuis plusieurs années de suite premier dans les sondages Stack Overflow concernant le langage le plus apprécié de ses utilisateurs. Les grandes entreprises semblent également être intéressées par son potentiel comme nous allons le voir dans un instant.

Pour faire un résumé de Rust, il s'agit d'un langage créé par Graydon Hoare en 2006, qui a ensuite été utilisé par la fondation Mozilla pour un nouveau projet avant que ce dernier soit suspendu. De nos jours, Rust est soutenu par la fondation Rust, une communauté qui contribue à améliorer le langage et qui est elle-même soutenue et financée par de grandes entreprises telles que Mozilla, Google, AWS, Microsoft ou encore Huawei.

Concernant la technologie en elle-même, Rust offre de nombreux avantages: c'est un langage compilé qui offre des performances équivalentes à C et C++, des langages bas niveau. De plus, sa gestion de la mémoire lui permet de d'éviter des erreurs de mémoires courantes que l'on peut retrouver dans les deux langages cités précédemment. La gestion de la mémoire n'est pas complètement manuelle à la manière de C ou C++, mais elle n'est pas automatique comme dans les langages haut niveau comme Python car Rust ne possède pas de ramasse-miettes. Rust utilise à la place des systèmes uniques tels que la propriété, l'emprunt et la durée de vie.

Rust possède aussi le concept de concurrence, ce qui permet à un programme d'effectuer plusieurs tâches en même temps et individuellement. Ainsi, Rust permet d'écrire du code performant tout en ayant une sécurité mémoire stricte, ce qui explique son intérêt. Par exemple, le kernel Windows qui a été en partie réécrit en Rust courant 2023 sur décision de Microsoft.

II) Le projet Servo

Servo est un moteur de rendu de navigateur web expérimental, initialement entré en développement en 2012 par Mozilla afin de créer une alternative plus moderne et performante au moteur Gecko utilisé par Firefox. Il s'agit du premier gros projet qui a utilisé le Rust car le créateur du langage, Graydon Hoare, était un ingénieur de Mozilla quand il a eu l'idée de créer le langage.

Servo tire parti des garanties de sécurité mémoire de Rust pour réduire les risques d'erreurs liés à la gestion de la mémoire, courants dans les moteurs de rendu C++. De plus, il exécute des tâches en parallèle, améliorant les performances sur les machines multi-cœurs. Ainsi, Servo peut rendre des pages plus rapidement en exploitant les processeurs modernes.

Cependant, le projet n'a pas encore abouti et Mozilla a réduit ses investissements, notamment à cause de la période Covid, où elle a suspendu le projet et licenciée son équipe travaillant sur le projet. La fondation Mozilla a décidé de mettre le projet en open source afin que la communauté grandissante de développeur Rust puissent contribuer. Les avancements réalisés sur ce projet ont été réutilisés dans d'autres projets de la fondation Mozilla, comme Quantum, qui a pour but de créer la nouvelle génération de moteur de recherche utilisée par la fondation.

III) Le Rust chez Discord

Discord étant un logiciel de messagerie instantanée populaire, il est nécessaire qu'il soit performant et sécurisé. Discord s'est mis en 2020 à réécrire en Rust son service « Statut de lecture », qui permet de définir si un message privé ou sur un serveur a été lu. Ce service gère des milliards de statuts.

L'entreprise avait créé ce service en utilisant Go à la base, mais s'est rendue compte que ce service connaissait des pics de latences importants parfois malgré de bonnes performances générales. Les développeurs ont compris que ces latences arrivaient à cause de la manière dont Go fonctionne au niveau de sa mémoire. Le problème venait du ramasse-miettes de Go, qui prenait trop de temps avant de libérer de la mémoire.

Les développeurs de Discord se sont alors mis à regarder du côté de Rust, ce qui a demandé aux développeurs d'utiliser une bibliothèque récemment créée et instable à ce moment là pour être en mesure de l'implémenter dans leur service. Une fois leur code réécrit, ils se sont rendus compte qu'ils ne subissaient plus de problèmes de latences, mais que leur code Rust non-optimisé était plus performant que leur code Go qui lui avait été optimisé auparavant dans le but de réduire les problèmes de latence cités précédemment.

IV) L'utilisation du Rust par Cloudflare pour ses outils

En tant qu'entreprise de sécurité et d'infrastructure réseau, Cloudflare doit traiter des millions de requêtes réseau par seconde, et une seule erreur comme un crash peut avoir de graves conséquences. Cela fait de Rust un langage adapté dans la situation de Cloudflare. Ainsi, le géant Américain a décidé d'utiliser Rust via la création de Pingora, un proxy HTTP créé car l'entreprise souhaitait remplacer son ancienne architecture proxy sous Nginx pour augmenter la taille de ses infrastructures.

Pingora est conçu pour être performant, notamment dans la gestion de connexions à grande échelle. Ce proxy arrive à traiter plus d'un trillion de requêtes par jour tout en fournissant de nouvelles fonctionnalités pour les utilisateurs, permet d'éviter les erreurs de gestion de mémoire type dépassement de tampon, et n'utilise que 1/3 des ressources mémoires et CPU qu'utilisaient l'ancienne architecture.

Cloudflare a également transformé Pingora en un framework Rust open source pour répandre l'utilisation du Rust dans la création d'outils réseaux, et ainsi améliorer la sécurité des infrastructures réseaux de manière générale.

V) Conclusion

Pour conclure, il est normal de voir que des grandes entreprises telles que des GAFAM s'intéressent à Rust et se mettent à créer des projets dans ce langage grâce à ses avantages de performances et de sécurité. On peut espérer à l'avenir que davantage d'entreprises se mettront à utiliser Rust pour créer des logiciels ou réécrire leurs programmes, notamment pour remplacer ceux écrits en C ou en C++. De plus, on peut noter que ces grandes entreprises ont mises pour la plupart d'entre elles leurs projets Rust en open source, probablement pour que la communauté de développeur Rust puisse contribuer à leur évolution et à l'évolution du langage.

VI) Sources

Afin de trouver les sources pour cet article, j'ai principalement regardé du côté des blogs des grosses entreprises pour voir différents projets réalisés en Rust, même si je n'ai pas parlé de tous les projets au final car les avantages de Rust sont les mêmes, ce qui serait devenu rapidement répétitif :

<https://discord.com/blog/why-discord-is-switching-from-go-to-rust>

<https://aws.amazon.com/blogs/aws/firecracker-lightweight-virtualization-for-serverless-computing/>

<https://medium.com/@Aaron0928/30-year-old-code-killed-microsoft-rewrites-windows-kernel-with-180-000-lines-of-rust-f891c95959f2>

<https://servo.org/>
<https://wiki.mozilla.org/Quantum>

<https://blog.cloudflare.com/tag/rust/>
<https://blog.cloudflare.com/how-we-built-pingora-the-proxy-that-connects-cloudflare-to-the-internet/>
<https://blog.cloudflare.com/pingora-open-source/>

<https://foundation.rust-lang.org/news/rust-x-google-summer-of-code-2024/>