

Hicham Khadda 4SI2
Racha Siad 4SI2

Documentation du Projet LoadBalancer

Description Générale

Ce projet implémente un load balancer simple en Rust en utilisant la crate hyper pour la gestion des requêtes HTTP et tokio pour l'asynchronicité. Le load balancer distribue les requêtes entrantes entre deux serveurs backend de manière cyclique et vérifie périodiquement leur état de santé.

Installation et Configuration

Pour utiliser ce projet, vous aurez besoin de Rust et de Cargo installés sur votre système. Si vous n'avez pas encore Rust, vous pouvez le télécharger et l'installer à partir de [<https://rustup.rs/>] (<https://rustup.rs/>).

Composants du Projet

Cargo.toml

Le fichier de configuration pour le projet Rust, définissant les dépendances nécessaires (hyper, tokio, lazy_static, etc.) et les binaires (loadbalancer, serveur1, serveur2).

src/main.rs

Le cœur du load balancer. Il initialise le serveur HTTP et utilise les fonctions définies dans backends.rs pour router les requêtes entrantes vers les serveurs backend appropriés. La fonction route_request modifie l'URI des requêtes pour les rediriger vers le backend sélectionné. Le serveur écoute sur le port 8080.

src/backends.rs

Contient la logique pour la sélection du backend (`select_backend`) et la vérification de l'état de santé des backends (`maintain_backends_health`). `select_backend` alterne les requêtes entre les deux backends, tandis que `maintain_backends_health` vérifie périodiquement leur accessibilité.

select_backend: Sélectionne de manière cyclique le prochain backend pour traiter une requête entrante.

maintain_backends_health: Vérifie l'état de santé des backends à intervalles réguliers.

src/serveur1.rs et src/serveur2.rs

Ces fichiers implémentent les serveurs backend simples qui répondent aux requêtes HTTP. Ils écoutent respectivement sur les ports 3000 et 3001 et renvoient une réponse de base pour confirmer leur fonctionnement.

Tests

Pour vérifier que le projet fonctionne exécuter cette commande :
`cargo test`

Pour compiler et exécuter le load balancer :
`cargo run --bin loadbalancer`

Pour lancer les serveurs backend :
`cargo run --bin serveur1`
`cargo run --bin serveur2`



Ces tests couvrent les fonctionnalités clés du load balancer, y compris :

- La sélection cyclique des backends.
 - La vérification de l'état de santé des backends.
 - La capacité du load balancer à router les requêtes entrantes vers le backend approprié.
- 