

Project DAAR

Mark-Compact GC for MINIZAM in C

Submitted by

Hejun Cao
Ruolin Zhou
Weida Liu

M1 Science et Technologie du Logiciel 2023-2024
Sorbonne Université (SU UPMC)

Under the guidance of

Bùi Xuân Bình Minh
Sorbonne Université (SU UPMC)

20/01/2025



Contents

1	Introduction	3
2	Architecture	3
3	Implémentation détaillée	3
3.1	Gestion des utilisateurs	3
3.2	KMP Search	5
3.2.1	Analyse	5
3.2.2	Structure de donnees	5
3.2.3	Implémentation	5
3.3	PageRank	5
3.3.1	Analyse	5
3.3.2	Structure de donnees	5
3.3.3	Implémentation	5
4	Tests et Résultats	5
4.1	Gestion des utilisateur	5
4.2	KMP Search	5
4.3	PageRank	5
4.4	Jmeter	5
5	Optimisation future	5
6	Conclusion	5

1 Introduction

Dans le cadre de notre projet DAAR, nous avons choisi de développer une application web/mobile de moteur de recherche pour une bibliothèque numérique, conformément aux exigences du "CHOIX A". Ce projet s'inscrit dans un contexte où les bases de données textuelles de grande envergure, comme la bibliothèque de Gutenberg, rendent la recherche manuelle inefficace. L'objectif principal est donc de concevoir une solution performante et intuitive permettant aux utilisateurs de localiser efficacement des documents textuels au sein d'une bibliothèque comprenant au minimum 1664 livres, chaque livre contenant au moins 10 000 mots.

Le projet s'articule autour de plusieurs objectifs clés :

- Un système de gestion des utilisateurs.
Pour gérer les recommandations personnalisées uniques de chaque utilisateur et le tri des résultats de recherche, ainsi que la fonction de favoris correspondante.
- KMP Search.
La recherche de contenu par expressions régulières à l'aide de KMP Search appris dans les cours de DAAR permet de trouver tous les contenus textuels de tous les livres et les résultats sont triés en fonction du nombre d'occurrences.
- Recommandation de contenu.
Nous utilisons l'algorithme PageRank pour calculer le score PageRank de chaque nœud (livre) dans le graphe de similarité des livres, les livres ayant un score élevé étant considérés comme les plus appréciés par l'utilisateur, tout en filtrant les résultats en fonction de l'auteur et de la catégorie afin de nous assurer que les livres recommandés correspondent aux intérêts de l'utilisateur.

En plus des défis techniques liés à l'implémentation des algorithmes de recherche et de classement, une attention particulière sera portée à la performance de l'application, mesurée à travers des tests sur des volumes de données conséquents.

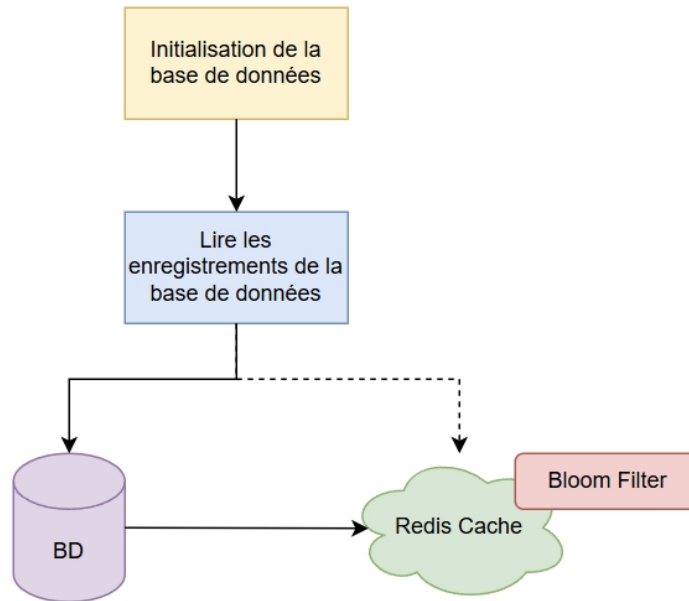
2 Architecture

3 Implémentation détaillée

3.1 Gestion des utilisateurs

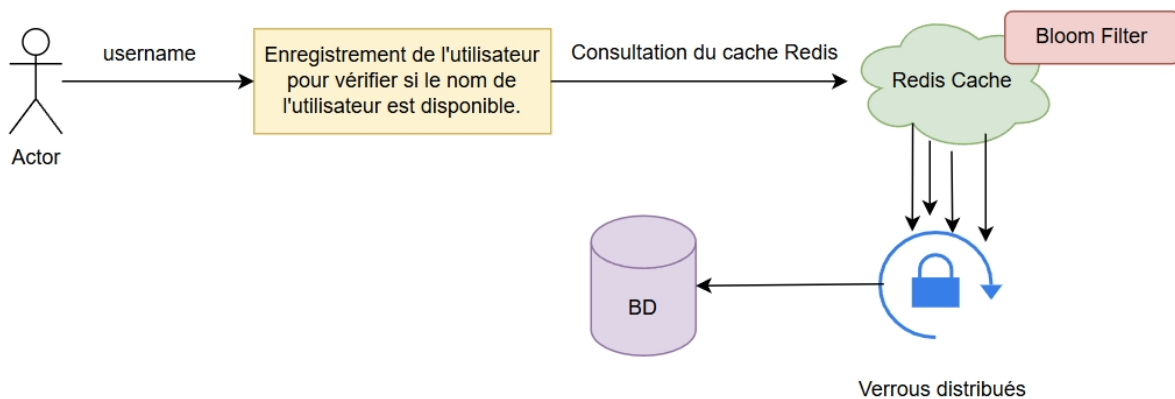
Notre module utilisateur met en œuvre les fonctions d'enregistrement, de connexion, de modification et de déconnexion des utilisateurs, ainsi que les fonctions de collecte de livres.

Nous avons mis en œuvre des modules utilisateurs pour la protection des bases de données à l'aide de filtres de Bloom, de verrous distribués, de buckets limitant les flux et de la mise en cache Redis.



À chaque fois que le projet s'exécute, nous lisons d'abord les données de la base de données, puis nous les synchronisons avec le cache Redis pour nous assurer qu'il n'y a pas de doublons.

Entre-temps, nous avons conçu les solutions suivantes pour faire face à d'éventuelles attaques de bases de données : S'il y a un grand nombre d'appels à l'interface d'enregistrement/de connexion avec le même nom d'utilisateur, nous utiliserons d'abord des filtres Bloom pour le filtrage, puis des verrous distribués pour verrouiller les noms d'utilisateur afin de garantir qu'une seule demande peut interroger la base de données en même temps.



En outre, nous avons mis en place une fonction de collecte facile. L'utilisateur crée un groupe et y place le livre correspondant. Les données du livre dans la collection affectent le poids de l'algorithme PageRank.

3.2	KMP Search
3.2.1	Analyse
3.2.2	Structure de donnees
3.2.3	Implémentation
3.3	PageRank
3.3.1	Analyse
3.3.2	Structure de donnees
3.3.3	Implémentation
4	Tests et Résultats
4.1	Gestion des utilisateur
4.2	KMP Search
4.3	PageRank
4.4	Jmeter
5	Optimisation future
6	Conclusion