Fonctionnement des Systèmes de Gestion de Bases de Données (FSGBD)

MASTER 1 MIAGE

DOSSIER TD DE GROUPE

PROJET DE DEVELOPPEMENT - ARBRE B PLUS

2023/2024

FEVRIER 2024

L'Unité d'Enseignement : Fonctionnement des Systèmes de Gestion de Bases de Données (FSGBD)

Enseignant : Monsieur Gregory Galli

Thème du sujet : Arbre B plus Spring

Étudiants : DAO Tuan Linh, TRINH Thi Thanh Thuy, Balrhi Anas et bocoum Allaye

DOSSIER DU TD

Table des matières

A. Création des données avec 100000 entrées

Pour créer le jeu de données avec 100000 entrées, nous avons utilisé Python pour créer un fichier et l'avons sauvegardé sous format CSV. Les données contiennent le numéro pour chaque clé, et leurs valeurs. Ces valeurs sont les noms des personnes.

241134	Tiffany Roberts
88065	Sharon Guzman
259410	Melissa Harper
613687	Michael Woodard
734461	Emily Martin
399783	Chase Taylor
430402	Kenneth Smith
27294	Lynn Johnson
889916	Jeanne Cook

Figure 1 : L'apparence des données

B. La recherche benchmark

1. Importation des données

Nous avons écrit une méthode nommée "importDataFromCSV" nous permettant d'importer le fichier vers l'arbre B plus.

2. Le résultat de la recherche benchmark

a) Moyen pour réaliser et le but

Le résultat est exporté dans le fichier nommé “benchmark_results.csv”. Ensuite, nous avons travaillé avec le logiciel RStudio pour extraire les informations statistiques et notamment comparer les deux méthodes : la recherche dans B arbre et la recherche séquentielle. Ici, le temps de recherche pour deux méthodes est compté par nano seconds.

Les données importées sous RStudio ont trois colonnes :

+ Première colonne contient les clés choisies
+ Deuxième colonne contient le temps de la recherche dans B arbre pour chaque clé
+ Troisième colonne contient le temps de la recherche séquentielle dans B arbre

Key <chr>	B.Tree.Search.Time..ns. <int>	Sequential.Search.Time..ns. <int>
729367	14500	16131600
872889	7200	22844100
310530	7100	21765100
551409	6000	47450000
538656	1900	14935400
532489	3400	20332400
630995	5600	5375900
282747	3700	2404900
211663	3500	861900
92136	4700	39365800

Figure 2 : Les données du temps de recherche sous RStudio

b) Les résultats statistiques

b.1) Le temps de la recherche sur les chiffres clés

La table de comparaison de temps de recherche entre deux méthodes :

Indicateurs statistiques	La recherche dans B Arbre (nanosecondes)	La recherche séquentielle (nanosecondes)
Min	400	217800
Moyenne	1877.6	19252972
Max	19100	58103900

Interprétation :
Selon le résultat ci-dessous, nous pouvons voir que le temps de la méthode “Recherche dans le B-tree” est généralement plus rapide et efficace dans ce cas que celui de la méthode “Recherche séquentielle” car le B-tree est une structure de données optimale pour la recherche. Voici quelques raisons principales:

1. **Organisation des données:** Le B-tree est conçu pour organiser les données de manière structurée et optimisée pour la recherche. Le B-tree divise les données en nœuds et fournit des index pour une recherche rapide, réduisant ainsi le nombre d'accès aux données.
2. **Distribution des données:** Dans un B-tree, les données sont distribuées sur plusieurs niveaux de l'arbre, du nœud racine aux feuilles. Ainsi, la recherche ne nécessite qu'un nombre relativement faible d'accès à l'arbre (généralement logarithmique du nombre de clés), tandis que la recherche séquentielle nécessite de parcourir chaque élément de la liste ou de l'ensemble.
3. **Équilibre de l'arbre:** Le B-tree est équilibré de manière efficace, garantissant que chaque nœud dans l'arbre contient un nombre relativement équilibré d'éléments. Cela se traduit par une répartition uniforme du temps de recherche sur tout l'arbre, évitant ainsi le pire des cas où un grand nombre d'éléments doit être parcouru, comme dans une recherche séquentielle.
4. **Complexité temporelle:** Le temps de recherche dans un B-tree peut être évalué à O(log n), tandis que la recherche séquentielle a généralement une complexité temporelle de O(n), où n est le nombre d'éléments dans l'ensemble de données. Ainsi, pour un grand ensemble de données, le temps de recherche dans un B-tree augmentera plus lentement que celui de la recherche séquentielle.

b.2) Le temps de la recherche présenté graphiquement

* Interprétation de la performance de la recherche B arbre

Ici, le temps de recherche varie entre 0 et 20000 nanosecondes. Pour la majorité des clés, le temps de recherche est inférieur ou égal à 5000 secondes, il y a juste quelques clés dont le temps de recherche est entre 10000 et presque 20000 nanosecondes.

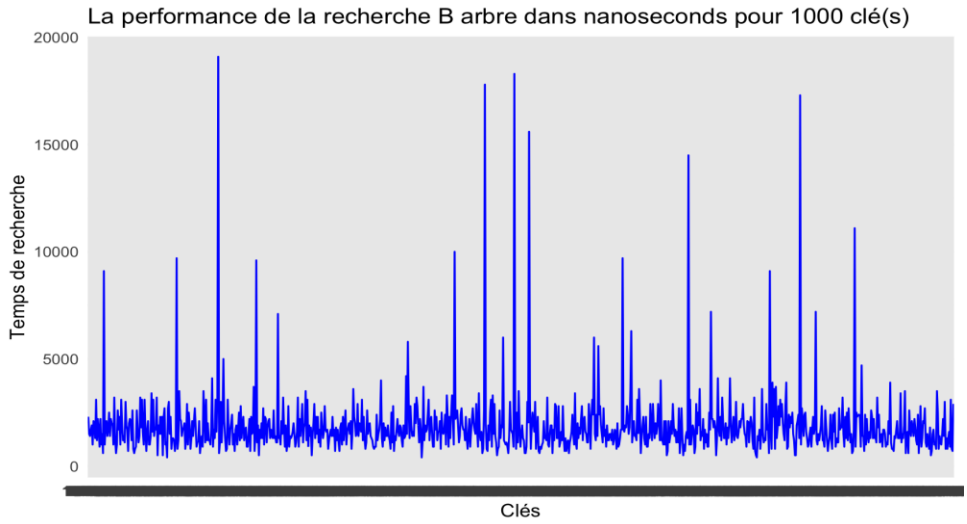

Figure 3: Diagramme de la performance de la recherche dans B arbre

* Interprétation de la performance de la recherche séquentielle

Ici, le temps de recherche varie entre 0 et 60000000 nanosecondes. Il y plus de variation par rapport au diagramme de la recherche dans B arbre. Pour la majorité des clés, le temps de recherche est entre 0 et 20000000 secondes. Nous pouvons voir que le temps de la recherche séquentielle est significativement plus grand que celui de la recherche dans B arbre.

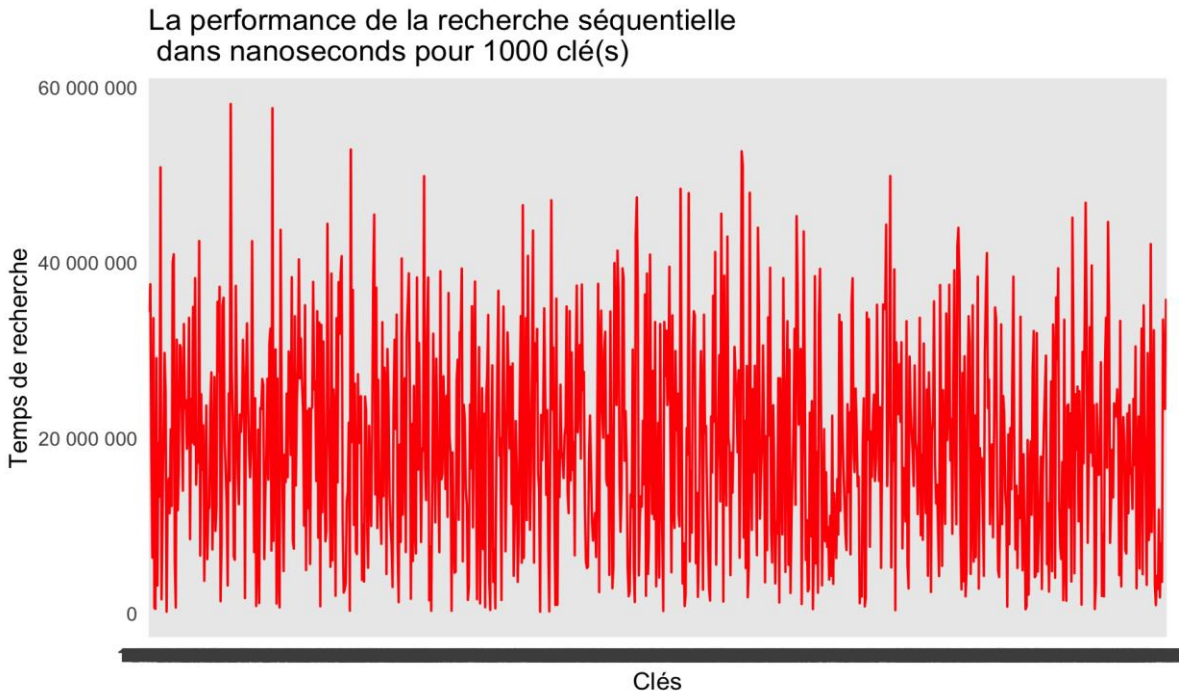

Figure 4* : *Diagramme de la performance de la recherche séquentielle