Rapport de tp 2 :

L’insertion standare et le bulk loading

INTRODUCTION :

Les B+Arbres sont des B-arbres équilibrés dont tous les feuilles sont au même niveau et contiennent tous les clés du fichier de donnée. Ces arbres sont très utiles pour la recherche d’une clé quelconque dans le fichier de donnée puisqu’il suffit de prendre une seule branche de l’arbre pour savoir si la clé existe ou non.

Dans ce TP on s’est intéressé au chargement initial de ces arbres qui devient très important lorsque fichier de donnée est volumineux.

Pour cela on a étudié deux méthodes différentes du chargement initial du B+arbre :le chargement à insertion standard et le Bulk Loading.

Dans la suite de ce rapport on va présenter des statistiques obtenues après l’exécution des deux algorithmes sur le même fichier de donnée ainsi que leurs représentations graphiques accompagnés d’interprétations et une comparaison des deux méthodes.

On a aussi utilisé deux versions différentes de la méthode d’insertion standard : l’une est avec la recherche de la position par contre l’autre est sans recherche car on connait déjà la position ou doit être insérer la valeur (car le fichier est ordonné) ;

Les Graphes :

1. **Temps en fonction du Nombre de Valeurs**

Le graphe montre la grande différence entre les deux méthodes en terme de performance. Comme on peut voir dans le graphe la création de l’arbre en utilisant le Bulk loading est beaucoup plus rapide que la méthode de l’insertion standard(avec Recherche) et plus le nombre de valeurs à insérer augmente plus l’écart entre les deux courbes devient important. On voit nettement que le Bulk Loading est meilleur que la méthode d’insertion standard car :

* Le fichier de donnée est préalablement trié qui a permis au Bulk Loading d’éviter le tri du fichier de donnée et qui peut prendre un temps important.
* Dans le Bulk Loading seule la première valeur de chaque feuille est inséré dans l’arbre ce qui réduit considérablement le nombre d’insertion donc le nombre d’accès mémoires (qui est très couteux) contrairement à la méthode de l’insertion standard qui sollicite l’aide du module de recherche à chaque insertion ce qui rend le temps de création important.

1. **Nombre de Nœuds ajoutés, Taille de l’arbre :**

On voit dans le graphe le nombre de nœuds ajoutés dans la méthode d’insertion standard est nettement plus grand que celle du Bulk Loading car comme expliqué précédemment , dans l’insertion en vrac ,on insère que les premières valeurs de chaque feuilles contrairement à l’autre méthode ce qui explique le grand écart qui devient énorme lorsque le fichier de donnée devient volumineux.

L’augmentation du nombre de nœuds implique l’augmentation de la taille de l’arbre.

1. Profondeur de l’arbre :

Pour les nombres de valeurs qu’on a pris La profondeur des deux arbres est presque la même (différence d’un degré) qui est très petite ce qui améliore la recherche des données par exemple pour rechercher une donnée dans un fichier de donnée qui contient plus de 25millions de valeurs il suffit de faire 3accés mémoire.

1. Nombre d’accès mémoire :

On voit nettement que le nombre d’accès mémoire lors de l’insertion standard(avec ou sans recherche) est beaucoup plus grand que celle du Bulk Loading(pour 100millions de valeurs il reste inférieur à 1m accés) . Car dans la première méthode (avec recherche) pour chaque valeur à insérer il faut vérifier qu’elle n’existe pas dans l’arbre et trouver l’emplacement où va être inséré, ce qui implique l’augmentation du nombre d’accès mémoire Contrairement au Bulk Loading la valeur à insérer est toujours insérer dans ce qu’on appelle la racine virtuelle qui est toujours en mémoire centrale tant qu’elle n’est pas pleine donc la valeur est insérer sans faire la recherche ce qui réduit considérablement le nombre d’accès mémoire. le nombre d’accès sans faire de la recherche affirme que le module de recherche est la raison pour laquelle le nombre d’accès est très grand dans la première méthode.

1. Nombre d’éclatements :

Comme le nombre de nœuds ajoutés dans la première méthode est plus grand que celle du Bulk Loading donc le nombre d’éclatements dans l’insertion standard serait systématiquement plus grand que celui du Bulk Loading.

Conclusion :

D’après les statistiques précédents on va clairement que plus le fichier de données est volumineux plus la méthode du Bulk Loading devient intéressante voir indispensable pour créer l’arbre initial puisque on voit nettement sa supériorité en terme de performances. Donc pour optimiser le temps utilise pour charger une base de données de taille considérable l’insertion en vrac est le choix idéal.

Mais malgré tous ses avantages l’insertion standard reste indispensable car si on veut insérer des valeurs après la création de l’arbre la méthode du bulk loading devient inutilisable.

TP réalisé par :

* Brahim Yacine G10
* Djellal Ahmed Abderrahmane G10