C018SA-W3-S5

Semaine 3 : Exécution et optimisation

1. Introduction
2. Réécriture algébrique
3. Opérateurs
4. Plans d'exécution
5. **Tri et hachage**
6. Algorithmes de jointure
7. Optimisation

Philippe Rigaux

Inria informatique mathématiques

Tri externe

Le **tri externe** est utilisé,

- pour les algorithmes de jointure (*sort/merge*)
- l'élimination des doublons (clause `distinct`)
- pour les opérations de regroupement (`group by`)
- ... et bien sûr pour les `order by`

Opération coûteuse sur de grands jeux de données.

Algorithme de **tri-fusion** (externe).

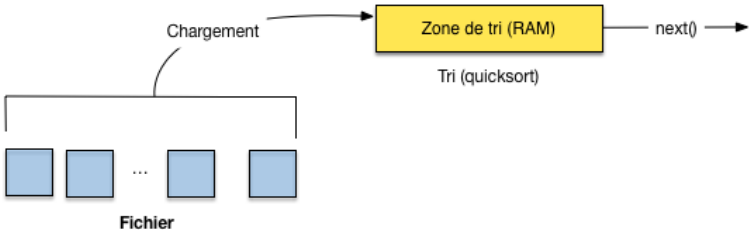
Première phase : le tri

Il faut une zone de tri, en RAM, allouée par le système.

On veut trier une source de données (ici, un fichier).

Première phase : le tri

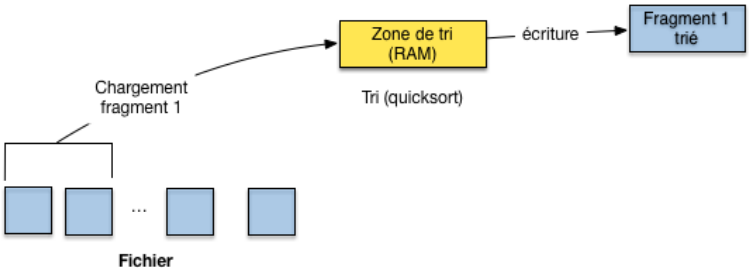
Cas favorable : tout le fichier tient en mémoire.



On charge, on trie avec *quicksort* : prêt pour l'itération.

Première phase : le tri

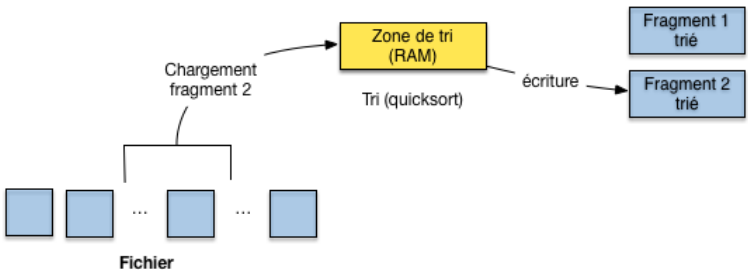
Et si le fichier ne tient pas en mémoire ?



On lit le premier fragment, on trie, on stocke.

Première phase : le tri

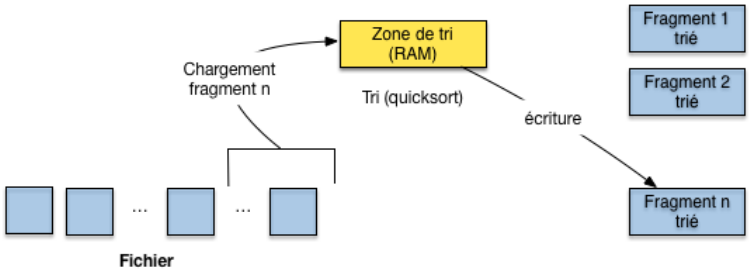
On lit le fragment suivant, on trie, on stocke.



Chaque fragment est **trié**.

Première phase : le tri

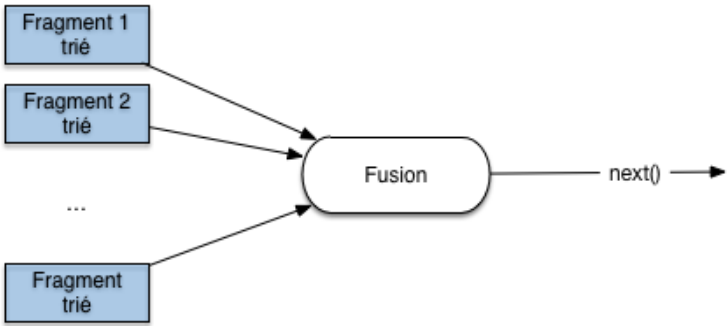
Production du dernier fragment trié



Prêt pour la phase suivante.

Fusion de listes triées

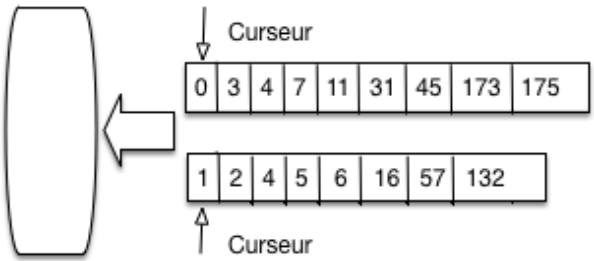
À l'issue de la phase de tri : n fragments triés.



Il faut maintenant les **fusionner**.

Fusion de listes triées

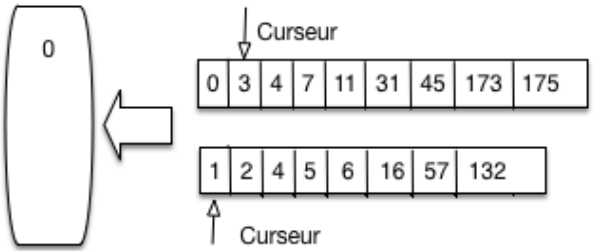
On place un curseur au début de chaque liste.



On compare les valeurs, et on prend la plus petite.

Fusion de listes triées

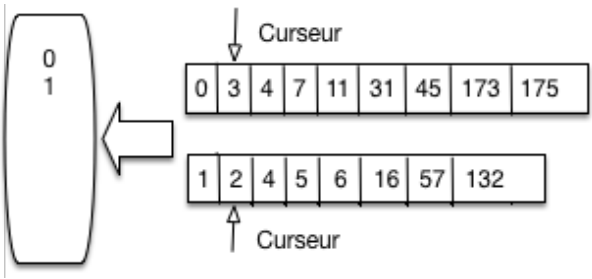
On a avancé sur le curseur de la valeur extraite.



On applique la même méthode.

Fusion de listes triées

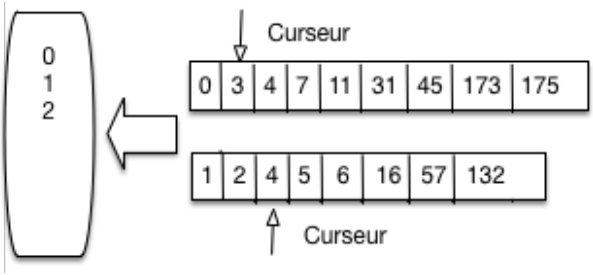
On continue sur le même principe.



On ne revient jamais en arrière.

Fusion de listes triées

Une dernière fois...



Le résultat est la liste triée globale.

Résumé : tri et opérateurs bloquants

L'opérateur de tri est **bloquant**

- La phase de tri est effectuée pendant le *open()*
- Le *next()* correspond à la progression de l'étape de fusion.

Conséquence : **latence importante des requêtes impliquant un tri.**

- Il faut lire **au moins une fois** toute la table.
- Si mémoire insuffisante, il faut lire deux fois, écrire une fois.
- Parfois pire...

Résumé : tri et opérateurs bloquants

L'opérateur de tri est **bloquant**

- La phase de tri est effectuée pendant le *open()*
- Le *next()* correspond à la progression de l'étape de fusion.

Conséquence : **latence importante des requêtes impliquant un tri.**

- Il faut lire **au moins une fois** toute la table.
- Si mémoire insuffisante, il faut lire deux fois, écrire une fois.
- Parfois pire...

Merci !