

Concepts Avancés de Bases de données

Joaquim LEFRANC et Jérôme SKODA

November 27, 2017

1 Nouveauté depuis le TP6

- Make gen-tp7 : Génération du fichier R du tp7 (necessite python3)
- Make demo-tp7 : Démonstration pour le TP7
- Ajout de diskManager (Writer ou Reader) pour la gestion des partition de disque
- Prise en compte de l'ordre alphabétique dans diskManagerReader et diskReader
- Uniformisation des méthodes d'écriture/lecture sur disques
 - diskManagerReader : Lecture de partition de disque
 - diskManagerWriter : Ecriture de partition de disque
 - diskReader : Lecture de disques
 - diskWriter : Ecriture de disque
- Ajout de methode de dump sur les diskReader et diskManagerReader
 - void disk_manager_r_dump(FILE* f, struct diskManagerReader* dmr);
 - void disk_r_dump(FILE* f, struct diskReader* dr);
- Ajout de test

2 Comment nous avons perdu betement 2 heures de notre vie

Lors de ce tp nous avons eu besoin de prendre en compte l'ordre des fichiers et nous avons donc choisi d'utiliser la fonction scandir. La fonction scandir avec alphasort permet d'obtenir le nom des fichiers contenu dans un dossier dans l'ordre alphabétique.

Exemple: 0.txt 1.txt 2.txt 3.txt 4.txt 5.txt 6.txt 7.txt etc.

Cependant, nous n'avons pas prévu le cas d'un merge avec plus de 10 fichier dans un bloque... Ce qui donne: 0.txt 1.txt **10.txt** 2.txt 3.txt etc.

Du coups, nous avons un algorithme qui fonctionnais dans les premiers steps puis dégénère dans les derniers steps.

Pour résoudre le problème, il faut ajouter remplacer "%d" par "%03d" dans le generateur de nom de diskManagerWriter et diskWriter. De cette manière les

fichiers seront nommés: 000.txt 001.txt 003.txt ... 009.txt 010.txt 011.txt Ce qui est drôle: un fix de 2 caractères pour 2 heures passées. Niveau de production exceptionnel!

3 Comment compiler le projet

- make all : Compile tout les fichiers
- make test : Lancement de la série de tests automatiques
- make doc : Génération de la documentation (doxygen)
- make rapport : Génération du rapport (latex)
- make clean : Nettoyage du projet (suppression des objets et binaires)
- make demo-tp1 : Lancer la démo tp1
- make demo-tp2 : Lancer la démo tp2
- make demo-tp3 : Lancer la démo tp3
- make demo-tp4 : Lancer la démo tp4
- make demo-tp5 : Lancer la démo tp5
- make demo-tp6 : Lancer la démo tp6
- make demo-tp7 : Lancer la démo tp7 (Nouveau)
- make rm-rs : Supprime le fichier res/RS.txt et res/disk/RS.txt
- make gen-tp5 : Génération de R pour tp5
- make gen-tp6 : Génération de R et S pour tp5
- make gen-tp7 : Génération de R pour tp7 (Nouveau)

4 Arborescence

- bin : Binaire exécutable
 - demo : Exécutable de démonstration
 - test : Exécutable de test
- doc : Documentation doxygen sous différents formats
- rapport : Source du rapport
- res : Ressources nécessaires au projet (fichier de bdd)
- script : Script utilisé pour les tests
- src : Source du projet
 - bdd : Source de la bibliothèque

- demo : Sources des differentes démonstrations d’utilisation
- test : Sources des dufferents tests
- sujet.pdf : Sujet du projet
- README.md : Le readme du projet
- rappot.pdf : C’est moi
- refman.pdf : Documentation format pdf

5 Caracteristiques

- Le code est organisé
- Il y a des code des tests
- Il y a la doc
- Il y a un rapport
- Et il y a pleins d’autre chose

6 Démonstration

Les sources de demosntration sont diponible dans: src/demo Les exécutable de test sont généré dans: bin/demo La commande make pour lancer les demo sont: make demo-tp1, make demo-tp2, make demo-tp3 etc...

- tp1-natural-join: Natural join R et S
- tp2-merge-join-without-duplicate: Merge join sans duplication
- tp3-merge-join-with-duplicate: Merge join avec duplication
- tp4-hash-join : Hash join
- tp5-nested-loop-disk: Nested loop sur disque
- tp6 nested loop join + hash join
- tp7-disk-sort-merge (Nouveau)

7 Test unitaire

Les sources de test sont diponible dans: src/test Les exécutable de test sont généré dans: bin/test Le script de test est dans script/test.sh La commande make pour lancer les test est: make test

- 00-storeFileBuffer: Ecriture d’un buffer dans un fichier
- 01-natural-join-1: Natural join R et S
- 02-natural-join-2: Natural join S et R

- 03-buf-quick-sort: Fonction de trie d'un buffer
- 04-merge-join-without-duplicate-1: Merge join sans duplication R et S
- 05-merge-join-without-duplicate-2: Merge join sans duplication S et R
- 06-merge-join-with-duplicate-1: Merge join avec duplication R et S
- 07-merge-join-with-duplicate-2: Merge join avec duplication S et R
- 08-hash-put-equilibre : Test ajout équilibré dans une table de hash
- 09-hash-put-desequilibre : Test ajout déséquilibré dans une table de hash
- 10-hash-full : Test remplissage complet dans une table de hash
- 11-hash-get : Test recuperation d'une entrée dans la table de hash
- 12-hash-remove : Test suppression / rehash dans une table de hash
- 13-hash-join : Test hash join
- 14-buffer-read-file: test la lecture avec un buffer extended
- 15-buffer-read-file-2: test la lecture avec un buffer extended
- 16-disk-buffer-dump: test la lecture d'un disque
- 17-disk-nested-loop-r-to-s-test: test le nested loop sur disque r to s
- 18-disk-nested-loop-s-to-r-test: test le nested loop sur disque s to r
- 19-buffer-decimal (nouveau)
- 20-table-puts (nouveau)
- 21-disk-block-hash-join (nouveau)

```

.../Documents/Master2/CavBDM2/tp6 7cc2a71 master
> make test
Script de test automatisé
[PASS] 00-storeFileBuffer
[PASS] 01-natural-join-r-s
[PASS] 02-natural-join-s-r
[PASS] 03-merge-join-without-duplicate-1-r-s
[PASS] 04-merge-join-without-duplicate-2-s-r
[PASS] 05-merge-join-with-duplicate-1-r-s
[PASS] 06-merge-join-with-duplicate-2-s-r
[PASS] 07-hash-remove
[PASS] 08-hash-put-equilibre
[PASS] 09-hash-put-desequilibre
[PASS] 10-hash-full
[PASS] 11-hash-get
[PASS] 12-hash-join-s-r
[PASS] 13-hash-join-r-s
[PASS] 14-buffer-read-file
[PASS] 15-buffer-read-file-2
[PASS] 16-disk-buffer-dump
[PASS] 17-disk-nested-loop-r-to-s-test
[PASS] 18-disk-nested-loop-s-to-r-test
[PASS] 19-buffer-decimal
[PASS] 20-table-puts
[PASS] 21-disk-block-hash-join
Script de test terminé

```