

# Concepts Avancés de Bases de données

Joaquim LEFRANC et Jérôme SKODA

November 19, 2017

## 1 Nouveauté depuis le TP5

- bufferExtended supporte maintenant les nombres (Mode BUFFER\_DECIMAL)
- Suppression du vieux buffer (toutes ces fonctionnalités sont incluse dans bufferExtended)
- Nouvelle structure: DiskOutput (gestion des ecriture sur disque)
- Nouvelle structure: Table (table de Hash sur disque)
- Nouvelle structure: Bucket (bucket de table)
- Ajout de fonction de benchmark Lecture/ecriture dans Buffer et Table
- Nouveau tests
- Nouvelle démo
- Nouvelle commande make gen-tp5 et gen-tp6 (pour generer R et S)
- Ajout de couleur pour les tests
- Mise à jour de make rm-rs
- Nouveau scripts (file-generation-tp6.py)

## 2 Comparaison de Disque Nested loop join avec Disk Block Hash Join

La comparaison s'effectue sur les même disque R et S contenant chacun 256 short. Les buffers sont de taille 10 supportant des données de taille décimales sur 2 octets. La table de hash a 10 bucket avec un hash linaire.

### 2.1 Hash join

- Création de table:
  - Disque R: Lecture de 256 lignes sur 26 fichiers
  - Disque S: Lecture de 256 lignes sur 26 fichiers
  - Table R: Ecriture de 256 lines sur 29 fichiers

- Table S: Ecriture de 256 lignes sur 32 fichiers
- Total opération: 1024 ligne sur 113 fichiers
- Jointure:
  - Table R: Lecture de 256 lignes sur 29 fichiers
  - Table S: Lecture de 594 lignes sur 73 fichiers
  - Disque RS: Ecriture de 32 lignes sur 4 fichiers
  - Total opération: 882 ligne sur 106 fichiers
- Total opération (creation table + jointure): 1906 ligne sur 219 fichiers

Remarque: Les opérations de lecture sur Table de S lors de la jointure peuvent être minimisé jusqu'à 256 ligne lecture s'il n'y a pas de collision (implique de mettre plus de bucket).

## 2.2 Nested Loop join

- Disque R: Lecture de 256 lignes sur 26 fichiers
- Disque S: Lecture de 6406 lignes sur 651 fichiers
- Disque RS: Ecriture de 32 lignes sur 4 fichiers
- Total d'opération: 6694 ligne sur 681 fichier

3 fois plus d'opération par rapport au hash join (creation de table + jointure).  
7,5 fois plus d'opération si nous comptons uniquement la jointure.

## 3 Comment compiler le projet

### 3.1 Avec le terminal

- 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 (Nouveau)

- make rm-rs : Supprime le fichier res/RS.txt et res/disk/RS.txt (Mis à jour)
- make gen-tp5 : Génération de R pour tp5 (Nouveau)
- make gen-tp6 : Génération de R et S pour tp5 (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 différentes démonstrations d'utilisation
  - test : Sources des différents tests
- sujet.pdf : Sujet du projet
- README.md : Le readme du projet
- rappot.pdf : C'est moi
- refman.pdf : Documentation format pdf

## 5 Caractéristiques

- Le code est organisé
- Il y a des codes des tests
- Il y a la doc
- Il y a un rapport
- Et il y a plein d'autres choses

## 6 Démonstration

Les sources de demonstration sont disponibles dans: `src/demo` Les exécutables de test sont générés dans: `bin/demo` La commande `make` pour lancer les tests est: `make demo-tp1, make demo-tp2, make demo-tp3` etc...

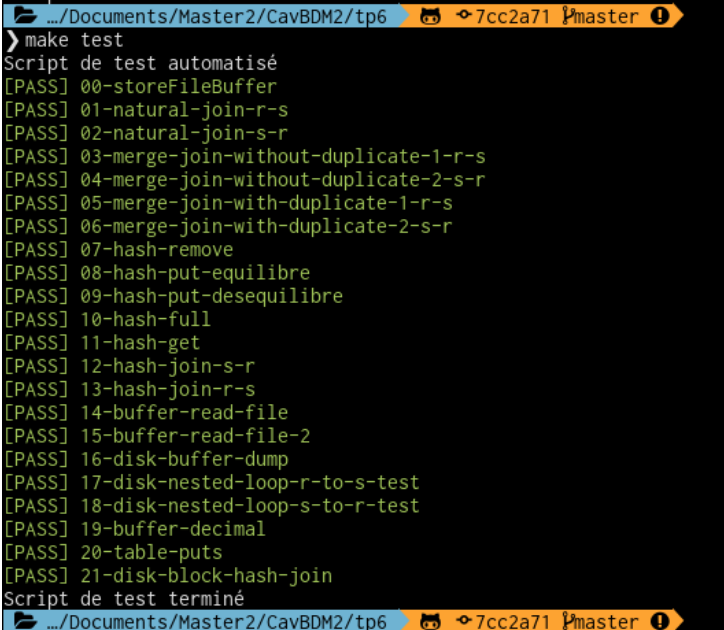
- `tp1-natural-join`: Natural join R et S
- `tp2-merge-join-without-duplicate`: Merge join sans duplication
- `tp3-merge-join-with-duplicate`: Merge join avec duplication
- `tp4-hash-join` : Hash join
- `tp5-nested-loop-disk`: Nested loop sur disque
- `tp6 nested loop join + hash join` (Nouveau)

## 7 Test unitaire

Les sources de test sont disponibles dans: `src/test` Les exécutables de test sont générés dans: `bin/test` Le script de test est dans `script/test.sh` La commande `make` pour lancer les tests 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 tri 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 récupération 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)

A terminal window with a dark background and light-colored text. The window title bar shows the path ".../Documents/Master2/CavBDM2/tp6" and a Git status bar indicating "7cc2a71" and "master". The terminal output shows a list of tests being executed, each preceded by "[PASS]". The tests listed are: 00-storeFileBuffer, 01-natural-join-r-s, 02-natural-join-s-r, 03-merge-join-without-duplicate-1-r-s, 04-merge-join-without-duplicate-2-s-r, 05-merge-join-with-duplicate-1-r-s, 06-merge-join-with-duplicate-2-s-r, 07-hash-remove, 08-hash-put-equilibre, 09-hash-put-desequilibre, 10-hash-full, 11-hash-get, 12-hash-join-s-r, 13-hash-join-r-s, 14-buffer-read-file, 15-buffer-read-file-2, 16-disk-buffer-dump, 17-disk-nested-loop-r-to-s-test, 18-disk-nested-loop-s-to-r-test, 19-buffer-decimal, 20-table-puts, and 21-disk-block-hash-join. The output ends with "Script de test terminé".

```
> 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
```