

Test de HSQLDB et Comparatif avec Sqlite

Table des matières

1 - Conditions préalables.....	2
2 - Installation de HSQLDB.....	2
3 - Premier Test de HSQLDB.....	2
4 - Deuxième Test pour bien comprendre :-).	2
5 - Configuration de OOo.....	2
5.a - Configuration du pilote JDBC dans OOo.....	2
5.b - Création de la source de données.....	3
6 - Tests effectués avec succès depuis OOo.....	3
7 - Problèmes rencontrés.....	3
8 - Remarques concernant l'installation sous Linux	3
8.a - Installation de Java.....	3
8.b - Remarque concernant « runManager.bat ».....	4
9 - Premières impressions.....	4
10 - Utilisation de HSQLDB en mode Client-Serveur.....	4
10.a - Configuration du serveur.....	4
10.b - Configuration du poste client et de OOo.....	4
10.c - Configuration de la source de données de OOo.....	5
10.d - Remarques concernant le mode Client-Serveur.....	5
10.e - Test des performances.....	5
10.f - Test de performances avec 4 tables liées.....	5
10.g - Comparatif performances avec 4 tables entre Sqlite, MySQL et HSQLDB.....	6
11 - Bilan des tests de HSQLDB.....	6
11.a - Points positifs.....	6
11.b - Points négatifs.....	6
11.c - Avis personnel.....	6
12 - Comparatif entre Sqlite et HSQLDB.....	7

1 - Conditions préalables

J'ai réalisé l'installation et les tests de HSQLDB sous Windows NT et sous Linux Debian avec Java j2re1.4.2_04 et OOo 1.1.2.

- Installer Java (Test effectué avec Java j2re1.4.2_04)
- Installer la dernière version de OOo avec prise en compte de Java (test effectué avec OOo 1.1.2)
- Vérifier que Java est opérationnel avec OOo (ex : Fichier / AutoPilote / État)

2 - Installation de HSQLDB

- Télécharger la dernière version sur le site <http://hsqldb.sourceforge.net> (Test effectué avec la version hsqldb_1_7_2_RC_6d). **ATTENTION : La dernière version stable hsqldb_1_7_1 ne fonctionne pas avec OOo.**
- Sur l'archive récupérée, il y a seulement deux fichiers intéressants
 - /hsqldb/lib/hsqldb.jar qui contient le programme, les utilitaires et le drivers JDBC
 - /hsqldb/demo/runManager.bat qui permet d'enregistrer la class java (c'est ce que j'ai compris, car je ne suis pas un spécialiste de java) et de lancer un utilitaire pour manager la base de données.

3 - Premier Test de HSQLDB

L'utilitaire « runManager.bat » permet de tester la base de données. Lancer ce programme et renseigner les champs suivants :

- Recent: Recent settings...
- Setting Name: test
- Type: HSQL Database Engine Standalone
- Driver: org.hsqldb.jdbcDriver
- URL:hsqldb:[file:test](#)
- User: sa
- Password:

Cet utilitaire, permet d'exécuter des commandes SQL pour gérer la base de données. Pour les réfractaires aux commandes SQL, il est possible d'utiliser le menu « Options / insert test data » pour créer 4 tables avec des données de test.

Par défaut la base de données est enregistrée dans le dossier « hsqldb/data » de l'archive, car la première ligne du programme « runManager.bat » contient « cd ..\data »

4 - Deuxième Test pour bien comprendre :-)

- Créer un nouveau dossier destiné à contenir la base de données (ex : C:\HSQLDB)
- Placer une copie des deux fichiers « hsqldb.jar » et « runManager.bat » dans ce dossier.
- Éditer le programme « runManager.bat » en supprimant la première ligne (« cd ..\data ») et en modifiant la deuxième ligne pour qu'elle corresponde à « @java -classpath hsqldb.jar org.hsqldb.util.DatabaseManager %1 %2 %3 %4 %5 %6 %7 %8 %9 »
- Lancer le programme en utilisant les mêmes paramètres que lors du premier test et utiliser le menu « Options / insert test data » pour créer des données de test.
- Avec ce nouveau test, la base de données est maintenant enregistrée dans le dossier de votre choix et le reste de l'archive téléchargée n'est plus nécessaire.

5 - Configuration de OOo

5.a - Configuration du pilote JDBC dans OOo

Pour configurer le pilote JDBC, OOo va avoir besoin du chemin du fichier « hsqldb.jar ». Il est

possible de placer ce fichier n'importe où, mais je pense que le meilleur emplacement est à l'intérieur du dossier de OOo (mais c'est un choix personnel). Donc pour suivre mon exemple, placer une copie du fichier « hsqldb.jar » dans le dossier d'installation de OOo (ex : C:\Program Files\OOo)

Indiquer à OOo le chemin du pilote JDBC :

- Menu « Outils / Options »
- Rubrique « OpenOffice.org / Sécurité »
- Dans Classpath, indiquer le chemin du fichier « hsqldb.jar » copié précédemment
 - ex : C:\Program Files\OOo\hsqldb.jar

5.b - Création de la source de données

Dans OOo, créer une nouvelle source de données et indiquer les paramètres suivants :

- Nom : HSQLDB
- Type de BD : JDBC et passer à l'onglet « JDBC »
- Class du pilote ODBC : org.hsqldb.jdbcDriver
- URL : hsqldb:C:\HSQLDB\test
 - (indiquez dans l'URL le chemin du dossier créé précédemment et le nom de la base de données)
- Nom d'utilisateur : SA

Pour tester si tout fonctionne, il est nécessaire de quitter OOo et le démarrage rapide de OOo

6 - Tests effectués avec succès depuis OOo

- Création / suppression de tables
- Ajout, modification, suppression d'enregistrements
- Création de sous-formulaires après avoir activé les paramètres nommés à l'aide de la macro
- Création de liens entre les tables depuis OOo (Intégrité référentielle). Là, je dois dire que j'ai été bluffé, car ni Sqlite, ni MySQL ne gère l'intégrité référentiel et HSQLDB la gère parfaite (Suppression et modification d'enregistrements en cascade...)

7 - Problèmes rencontrés

- Modification de table impossible (Commande ALTER TABLE non supporté comme Sqlite)
- La base de données est chargée entièrement en mémoire lors de la première utilisation, ce qui implique un temps d'attente assez long avec une base importante.
- L'utilitaire « runManager.bat » ne supporte pas les minuscules dans le nom des tables alors que OOo le supporte bien.
- L'utilitaire « runManager.bat » ne supporte pas d'être lancé depuis un chemin réseau, il faut obligatoirement créer un lecteur réseau pour pouvoir le lancer.
- Il n'est pas possible d'accéder à la base depuis plusieurs postes simultanément.

8 - Remarques concernant l'installation sous Linux

8.a - Installation de Java

- J'ai téléchargé sur le site <http://www.java.com/fr/download/manual.jsp> le binaire (pas le RPM) pour linux.
- J'ai placé ce binaire dans le dossier /opt (comme pour OOo)
- J'ai exécuté ce binaire qui m'a demandé la validation de la licence et qui a décompressé dans le répertoire courant Java.
- Ensuite, lors de l'installation de OOo, java n'est pas détecté automatiquement, mais il suffit juste de lui indiquer le dossier d'installation de java (ex : /opt/java)
- Si OOo à été installé avant java, il faut utiliser le programme « jvmsetup » situé dans le

dossier « program » de OOo pour configurer java dans OOo

8.b - Remarque concernant « runManager.bat »

Ce programme ne fonctionne pas sous Linux, pour le remplacer, il faut en ligne de commande, se placer dans le dossier contenant la base de données et le fichier « hsqldb.jar » et saisir la commande suivante :

➤ `java -classpath hsqldb.jar org.hsqldb.util.DatabaseManager`

9 - Premières impressions

- Étant donné que la base est entièrement chargée en mémoire, il n'est pas possible de l'utiliser pour de grosses bases de données.
- Cette base de données n'est pas utilisable par plusieurs utilisateurs simultanément, ce qui limite pas mal son utilisation (ce qui est logique étant donné qu'elle est chargée en mémoire)
- Il n'est pas possible comme avec Sqlite de modifier une table, ce qui est également très gênant pour une utilisation courante.
- A priori, je trouve que HSQLDB est moins buggué et plus facile à installer que Sqlite. Donc HSQLDB est plus adapté pour une utilisation mono utilisateur avec une petite base de données que Sqlite.

10 - Utilisation de HSQLDB en mode Client-Serveur

10.a - Configuration du serveur

- Installer Java sur le serveur (Test effectué avec Java j2re1.4.2_04)
 - Une fois l'archive décompressée, il suffit de copier le répertoire dans le répertoire de votre choix.
 - ex : `/var/j2re1.42_04`
- Télécharger la dernière version de HSQLDB sur le site <http://hsqldb.sourceforge.net> (Test effectué avec la version `hsqldb_1_7_2_RC_6d`). **ATTENTION : La dernière version stable `hsqldb_1_7_1` ne fonctionne pas avec OOo.**
- Sur l'archive récupérée, il n'y a que le fichier « `/hsqldb/lib/hsqldb.jar` » de nécessaire. Il contient le programme, les utilitaires et le drivers JDBC.
- Copier le fichier « `hsqldb.jar` » dans le dossier devant contenir la base de données (ex : `/var/hsqldb`)
- La commande « `/var/j2re1.4.2_04/bin/java -classpath hsqldb.jar org.hsqldb.util.DatabaseManager` » permet de lancer un utilitaire pour tester la base de données et enregistrer la class Java (C'est ce que j'ai compris...)
 - `/var/j2re1.4.2_04/bin/java ->` Indiquer le chemin de Java
 - `-classpath hsqldb.jar org.hsqldb.util.DatabaseManager ->` Indique qu'il faut lancer le Manager
- La commande « `/var/j2re1.4.2_04/bin/java -classpath hsqldb.jar org.hsqldb.WebServer -port 8080 -database testhsqldb` » permet de lancer le serveur.
 - `/var/j2re1.4.2_04/bin/java ->` Indiquer le chemin de Java
 - `-classpath hsqldb.jar org.hsqldb.WebServer ->` Indique qu'il faut lancer le serveur
 - `-port 8080 ->` Indique le port en écoute sur le serveur. Par défaut le port écouté est le 80, mais si le serveur est également utilisé en serveur Web, il faut utiliser un autre port.
 - `-database testhsqldb ->` Indique le nom de la base de données à utiliser. Si la base n'existe pas, elle sera créée dans le dossier contenant le fichier « `hsqldb.jar` »

10.b - Configuration du poste client et de OOo

- L'installation est la même que pour utiliser HSQLDB en local.
 - Installer Java

- Installer OOo avec prise en compte de Java
- Placer le fichier « hsqldb.jar » dans le dossier de votre choix (ex : /program/classes de OOo)
- Dans « Menu « Outils / Options » et rubrique « OpenOffice.org / Sécurité » et dans Classpath, indiquer le chemin du fichier « hsqldb.jar »

10.c - Configuration de la source de données de OOo

Par rapport à l'installation en locale, c'est la seule chose qui change.

- Nom de la source de données : ex : HSQLDB_Web
- Type de BD : JDBC et passer à l'onglet « JDBC »
- Class du pilote ODBC : org.hsqldb.jdbcDriver
- URL : hsqldb:http://NomDuServeur:8080 -> Mettre le numéro du port en écoute sur le serveur. Dans mon exemple j'ai utilisé 8080, car le port 80 est déjà utilisé par le serveur Web Apache.
- Nom d'utilisateur : SA

10.d - Remarques concernant le mode Client-Serveur

Le principal intérêt de ce mode est de pouvoir accéder à la base de données depuis plusieurs postes simultanément (Test effectué depuis un poste client sous Win NT et un autre sous Linux Debian).

10.e - Test des performances

J'ai réalisé un test pour vérifier si HSQLDB était performant au niveau rapidité et je dois dire que j'ai été un peu déçu mais pas très surpris, car Java n'est pas réputé pour ça vitesse....

Configuration du test :

- Poste client : Celeron 700 avec 256Mo de Ram
- Poste Serveur : XEON 1.0 Ghz avec 512Mo de Ram
- Table de 59000 enregistrements de 6 champs et non indexé

Pour réaliser les test, j'ai effectué un transfert des données de 4400 enregistrements de la table dans Calc et voici les résultats :

- HSQLDB en mode standard : 32,3s pour transférer les 4400 lignes dans Calc
- HSQLDB en mode Client-Serveur : 32,3s pour transférer les 4400 lignes dans Calc. Non, je n'ai pas triché, c'est exactement le même temps au dixième de seconde prêt... :-)
- MySQL en mode Client-Serveur via ODBC : 1,8s pour transférer les 4400 lignes dans Calc.
- MySQL en mode Client-Serveur via JDBC : 34,8s pour transférer les 4400 lignes dans Calc.
- Sqlite en local via ODBC : 1,7s pour transférer les 4400 lignes dans Calc.

Donc, MySQL est 18 fois plus rapide que HSQLDB pour réaliser cette opération..... :-(

Et MySQL est 19 fois plus rapide avec un driver ODBC qu'avec un driver JDBC..... :-(

Sqlite est aussi rapide que MySQL via ODBC

10.f - Test de performances avec 4 tables liées

Configuration du test :

- Poste : P2 350 avec 192Mo de Ram
- Requête sur les 4 tables de testes fournies avec HSQLDB :

Voici le résultat des tests avec HSQLDB

- Requête sur CUSTOMER (50 enregistrements) = <5s
- Requête sur CUSTOMER+INVOICE (50 enregistrements) = <5s
- Requête sur CUSTOMER+INVOICE+ITEM (650 enregistrements) = <5s
- Requête sur CUSTOMER+INVOICE+ITEM+PRODUCT (650 enregistrements) = **4mn 50s !!**

Avec 4 tables sous OOO, SQLDB a mis **4mn50s** pour afficher le résultat d'une requête de 650 enregistrements, ce qui est catastrophique... :-)

Mais en utilisant l'utilitaire livré avec SQLDB, le résultat était instantané. Donc le problème vient de OOO ou du driver JDBC.

10.g - Comparatif performances avec 4 tables entre Sqlite, MySQL et HSQLDB

Configuration du test :

- Poste : Celeron 700 avec 256Mo de Ram
- Requête sur les 4 tables de testes fournies avec HSQLDB (650 enregistrements retournés) :

Voici le résultat des tests :

- HSQLDB via OOO = 88,8s
- HSQLDB via l'utilitaire = <1s
- MySQL via ODBC = 0,4s
- MySQL via JDBC = 2,8s
- Sqlite via ODBC = 0,3s

Le temps d'exécution de la requête via OOO avec HSQLDB est désastreux et rend cette base inexploitable dans l'état actuel alors que Sqlite et aussi rapide que MySQL.

11 - Bilan des tests de HSQLDB

11.a - Points positifs

- Très facile à installer
- Installation identique sous Linux et Sous Windows (Sans tenir compte de l'installation de Java)
- Stockage de la base sous forme de requêtes SQL ce qui est très original et ouvre de nouvelles possibilités.
- Gestion des droits limité mais c'est déjà mieux qu'Access ou Sqlite.
- Gestion de l'intégrité référentiel ce qui est mieux que MySQL.
- Pas de bugs constaté.
- Plusieurs modes d'utilisation possible (En local ou en réseau). Le mode en réseau est le seul qui permet l'accès simultané par plusieurs personnes.

11.b - Points négatifs

- Étant donné que la base est entièrement chargée en mémoire, il n'est pas possible de l'utiliser pour de grosses bases de données.
- Cette base de données n'est pas utilisable par plusieurs utilisateurs simultanément (sauf en mode réseau), ce qui limite pas mal son utilisation.
- Il n'est pas possible via OOO de modifier une table.
- Impossible de faire fonctionner depuis OOO, les champs auto-incrémentés
- Pour transférer 4000 lignes dans Calc, cette base est 18 fois plus lente que Sqlite ou MySQL via ODBC. Cela est essentiellement dû à l'utilisation de Java.

11.c - Avis personnel

Si HSQLDB est destiné à être intégré à OOO dans le but de concurrencer Access (ce qui est une très bonne chose) je trouve qu'il est vraiment dommage qu'il ne soit pas possible d'accéder à la base de données au moins en lecture seule depuis plusieurs postes ce que fait Access très bien (Je ne parle pas de l'écriture simultané, mais seulement de la lecture).

12 - Comparatif entre Sqlite et HSQLDB

Je donne une note de 0 à 2 en fonction des résultats (0=Nul, 1=Moyen, 2=Bien)

Point de comparaison	Sqlite	Note	HSQLDB	Note
Multi-plateforme	Windows et Linux -> OK	2	Windows et Linux -> OK	2
Installation	Difficulté sous Debian car il faut recompiler UnixODBC, mais si Sqlite est intégré à OOo il n'y aura plus de problème	1	Très facile	2
ALTER TABLE	Sqlite ne supporte pas la commande SQL ALTER TABLE	0	HSQLDB supporte Alter Table mais pas sous OOo pour le moment	1
Auto-Incrément	OUI	2	Pas sous OOo	1
Nécessité d'installer Java	NON	2	OUI	0
Sous-Formulaire	OUI	2	Nécessité activation paramètres nommées	1
Gestion des droits	NON	0	Oui, mais limité	1
Intégrité référentielle	NON	0	Oui, mais pas indispensable pour une petite base	1
Multi-Utilisateurs	Pas en mode local	0	Oui, mais pas sous OOo	1
Évolution	Sqlite est intégré à PHP5, ce qui garantie une forte évolution	2	?	0
Performance 1	1,8s pour transférer 4400 lignes	2	32,3s pour transférer 4400 lignes	0
Performance 2	0,3 pour exécuter sous OOo une requête avec 4 tables	2	88,8s pour exécuter sous OOo une requête avec 4 tables	0
Nombre de fichiers	1	2	3 ou 4	1
TOTAL		17		11

Tony GALMICHE le 06/07/2004

Maj le 13/07/2004 -> Ajout Utilisation de HSQLDB en mode Client-Serveur

Maj le 15/07/2004 -> Ajout temps accès MySQL via JDBC et Sqlite via ODBC

Maj le 16/07/2004 -> Ajout Bilan des tests de HSQLDB

Maj le 17/07/2004 -> Ajout comparatif Sqlite – HSQLDB et Test de performances avec 4 tables liées

Maj le 19/07/2004 -> Test de performances avec 4 tables liées