

Les bases de l'optimisation SQL avec DB2 for i

Christian GRIERE
cgriere@fr.ibm.com
Common Romandie
3 mai 2011

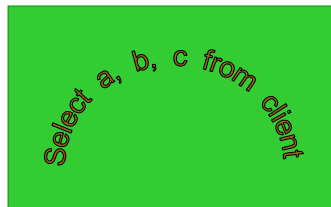
Les fleurs et les requêtes

Lorsque l'on veut planter de nouvelles fleurs dans un jardin
il faut :

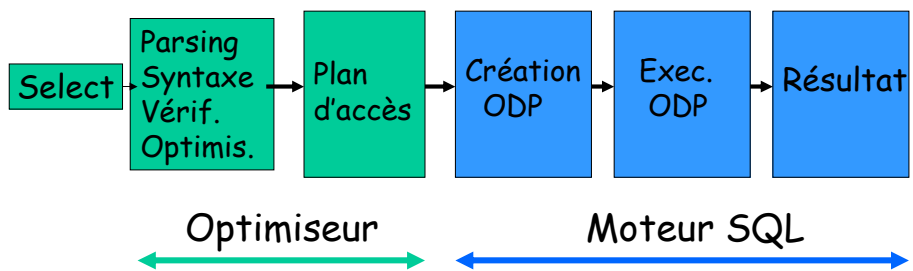
- choisir une bonne variété
- choisir un bon emplacement
- surveiller sa croissance
- la traiter (éventuellement)
- écrire une belle requête
- choisir un bon environnement
- surveiller son exécution
- l'optimiser (éventuellement)



=



La vie d'une instruction SQL



Les types de conseil

- Ecriture des instructions SQL (6)
- Environnement des instructions SQL (6)
- Optimisation des instructions SQL (8)

Conseils d'écriture des instructions SQL

N° 1 - Eviter les écritures dites de « facilité »

Exemple :

- `SELECT * FROM ...` mais plutôt :
`SELECT col1, col4, col6 FROM ...`
 - Diminue le nombre de calls faits au driver
Exemple : Une classe Java qui exécute 20 fois une instruction SQL qui interroge une table de 20 colonnes.
'SELECT * ' génère 800 calls database dans le driver
'SELECT col1, col4, col6' génère 120 calls database dans le driver
 - Augmente les chances d'utilisation d'un IOA (Index Only Access)
 - Diminue la charge réseau
-

N° 2 - Soyez causant ...

Donner à DB2 for i le maximum d'information concernant vos requêtes.

Clauses : FOR READ ONLY,
FOR UPDATE OF,
FETCH FIRST N ROWS,
OPTIMIZE FOR ALL/n ROWS

N° 3 - Surveiller les LIKE

Faire attention à l'opérateur LIKE lorsqu'il est associé à un paramètre du style %xxx%

N° 4 - Eviter les fonctions sur colonne

- Eviter les fonctions sur les colonnes de la clause WHERE avant la version 6.1

Exemple :

```
SELECT ... FROM CLIENT WHERE UPPER(nomcli) = '?'
```

- Autorisé à partir de la 6.1 grâce aux index de type « FBI » :

```
CREATE INDEX CLIENT_IX1 on CLIENT (UPPER(nomcli))
```

N° 5 - Proscrire l'utilisation des fichiers logiques

Ne pas utiliser de fichier logique comme nom de table dans les SELECT

N° 6 - Respecter le paradigme ...

- En SQL dynamique : 1 préparation avec marqueurs, N exécutions
 - Instruction SQL à un seul endroit dans le code
-

Conseils de bon environnement
pour les instructions SQL

N° 1 - Choisir l'une des dernières versions de l'IBM i

- Etre en IBM i 6.1 ou 7.1
 - Avoir installé le dernier niveau du groupe de PTF DB2 for i
-

N° 2 - Vérifier les valeurs système liées à DB2 for i

- QQRVDEGREE
 - *NONE = pas de // IO pour CQE (défaut)
// IO pour SQE
pas de // CPU pour CQE/SQE
 - *IO = // IO pour CQE/SQE
 - *OPTIMIZE = // IO ou // CPU pour CQE/SQE

NB : *OPTIMIZE requiert le dispositif DB2 SMP

- QDBFSTCCOL *ALL (défaut)
-

N° 3 - Disposer d'un pool mémoire partagé/privé

- Disposer d'un pool partagé/privé de mémoire pour les travaux serveur BD (QZDASOINIT, QSQSRVR, ...)
 - Si l'ajusteur de performance est actif lui fournir des consignes pour ce pool :
 - % taille minimale de mémoire
 - priorité
 - Si mémoire non contrainte activer l'expert cache pour ce pool
-

N° 4 - Réorganiser les réf. croisées de SQL

Réorganiser les fichiers des références croisées SQL si nécessaire

- Globalement :
 - RCLSTG SELECT(*DBXREF)
 - Si nécessaire :
 - RCLDBXREF OPTION(*CHECK) LIB(*ERR)
 - RCLDBXREF OPTION(*FIX) LIB(MaLib)
-

N° 5 - Réorganiser les fichiers/tables

- Réorganiser les fichiers/tables des applications pour supprimer les lignes supprimées (RGZPFM)

Attention aux produits de réplication logiciel

- Paramètre REUSEDLT(*NO/*YES)
 - Par défaut un fichier est en REUSEDLT(*NO)
 - Par défaut une table est en REUSEDLT(*YES)
-

N° 6 - Optimiser la journalisation

- Utiliser toutes les unités de disques disponibles pour les récepteurs de journaux (seuil de détachement > 64 Mo * Nb d'unités)
 - Séparer la journalisation de chemins d'accès et de données
RCVSIPOPT(*RMVINTENT)
 - Limiter les données à journaliser
 - Vérifier le temps attribué à SMAPP pour reconstruire les chemins d'accès en cas de fin anormale de la partition
 - Envisager l'option 42 de l'IBM i (HA Journal Performance)
-

Conseils d'optimisation des instructions SQL

N° 1 - Avoir une stratégie d'indexation de base

Qui évite :

- les créations de certaines structures temporaires (notamment les index)
 - les lectures séquentielles de table ou d'index ayant beaucoup de lignes
-

N° 2 - Chasser le CQE

Vérifier pourquoi CQE est encore utilisé pour certaines requêtes et réagir.

- Table avec logique select/omit sous-jacent ? *
- Conversion de type de donnée (upper, lower, CCSID) ? **
- Fichier logique ? ***

* : pris en charge à partir de la 5.4

** : pris en charge à partir de la 6.1

*** : pris en charge à partir de la 7.1

N° 2 - Chasser le CQE - Comment ?

Avec les 2 PTF SI40460 (6.1) et SI39207 (5.4) vous pouvez lancer un moniteur de performance SQL sur une plus longue période :

```
STRDBMON ... COMMENT('WANT_CQE_ONLY')
```

Attention aux INSERT simples

N° 3 - Eviter la prolifération des index

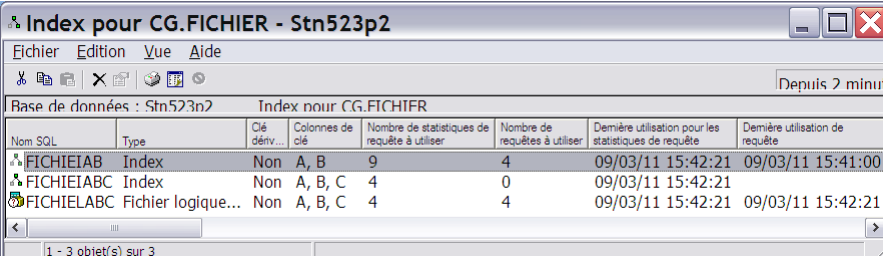
Création d'un index sur colonnes A, B, C
puis
création d'un index sur colonnes A, B
=
2 chemins d'accès à maintenir

NB : Avec les logiques il y avait partage du chemin d'accès

N° 4 - Supprimer les index inutilisés

Pour chaque structure d'accès il y a deux compteurs et deux dates :

- Compteur et date d'utilisation en phase d'optimisation (SQL)
- Compteur et date d'utilisation en phase d'exécution (SQL)



The screenshot shows a window titled 'Index pour CG.FICHIER - Stn523p2'. It contains a table with the following data:

Nom SQL	Type	Cle dériv.	Colonnes de clé	Nombre de statistiques de requête à utiliser	Nombre de requêtes à utiliser	Dernière utilisation pour les statistiques de requête	Dernière utilisation de requête
FICHIELAB	Index	Non	A, B	9	4	09/03/11 15:42:21	09/03/11 15:41:00
FICHIELABC	Index	Non	A, B, C	4	0	09/03/11 15:42:21	
FICHIELABC	Fichier logique...	Non	A, B, C	4	4	09/03/11 15:42:21	09/03/11 15:42:21

At the bottom of the window, it says '1 - 3 objet(s) sur 3'.

N° 5 - Etre observateur ...

Savoir distinguer une instruction SQL lourde optimisée
d'une instruction SQL légère non optimisée ...

5 000 exécutions à 1 s = 5 000 secondes

500 000 exécutions à 10 ms = 5 000 secondes

10 ms → 2 ms = 1 000 secondes

N° 6 - Avoir une stratégie d'indexation « haut de gamme »

Pour :

- les requêtes répétitives
- les requêtes lourdes

Ce sont des VIR ... à ne pas confondre avec des VIP

N° 6 - Avoir une stratégie d'indexation « haut de gamme »

La stratégie d'indexation « haut de gamme » recouvre :

- les index parfaits
- la réécriture automatique des requêtes
- l'utilisation d'index IOA
- le scannage d'index
- l'utilisation d'index EVI
- le GAP/LPG (Génération Anticipée de Prédicats/Look-ahead Predicate Generation)
- les jointures en étoile (star join/snow join)

N° 7 - Analyser les créations de MTI et les remplacer par des index permanents


 select dbname as "Schéma", sys_tname as "Table", table_size ...

Schéma	Table	Nb lig. table	Clés conseillées (SQE)	Nb MTI créés	Nb util. MTI	Date dern. util. du MTI
KBPPFR	QZG0001293	99860	QOKS, QQRID, QQILNM	1	1	1/2011-02-02 11:00:44.006179
CBPPFR	QZG0001293	99860	QQRID, QQSTIM	1	14	1/2011-02-02 10:58:33.435593
CG	DSPFFD	28568	WHFTYP	1	3	3/2011-01-20 14:28:06.699358
CG	QZG0000588	9477	QQRID, QQJFLD	1	10	10/2011-01-17 13:38:23.705647
FER1PFRP	MZG0000292	72664	QQRID, QQJFLD	1	2	2/2011-02-26 16:42:08.334398
FER1PFRP	BOUCLEM	558017	QQTIME	1	2	2/2011-02-26 10:53:14.792303

N° 8 - Envisager un achat ...

Envisager l'acquisition de l'option 26 de l'IBM i :
DB2 Symmetric Multiprocessing

▪ Intérêts :

- Choix d'un traitement // CPU des requêtes
- Maintenance en // des index en insertion groupée (même en programmation HLL)
- RGZPFM en // CPU

▪ Prérequis :

- Ressources (CPU, mémoire et bras disque)
-

Outils de performance
DB2 for i

Outils performance pour DB2 for i

- Ils sont tous standards dans l'IBM i et tous ... graphiques
 - Moniteur de performance SQL
 - Cache de plan
 - Visual Explain
 - Index Advisor
 - Index Condensor

Moniteur de performance SQL

Vue globale d'analyse pour FH - Global - Base - Stn523p1(Stn523p1)

Fichier Actions Aide

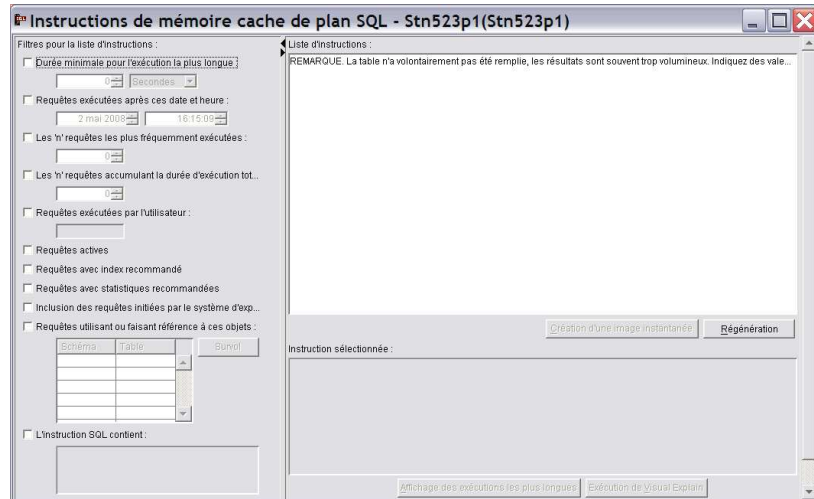
Données récapitulatives

	Valeur	Récapitulatif disponible	Instructions disponibles
Overview			
• SQL Statements	33584	✓	✓
• Users	12	✓	
• Jobs	200	✓	
• Threads	200		
• Average Table Rows	119759,709		
• Average Rows Returned	129,256		
• Average Runtime	0,009116		
• Average Parallel Degree Used	1		
• Maximum Parallel Degree	1		
• SQE	13712	✓	✓
• CQE	4570	✓	✓
• System Naming	28270	✓	✓
• SQL Naming	408	✓	✓
• Unique Open Statements	196	✓	✓
• Full Opens	4648	✓	✓
• Pseudo Opens	14831	✓	✓
• Average MQTs Used	0	✓	✓
• Average Indexes Used	0,966	✓	✓
• Full Indexes Created	267	✓	✓
• Sparse Indexes Created	3	✓	✓
• Index From Index Created	43	✓	✓
• Index Creates Advised	896	✓	✓
• Advised Statistics	23174	✓	✓
• Temporary Tables	5	✓	✓

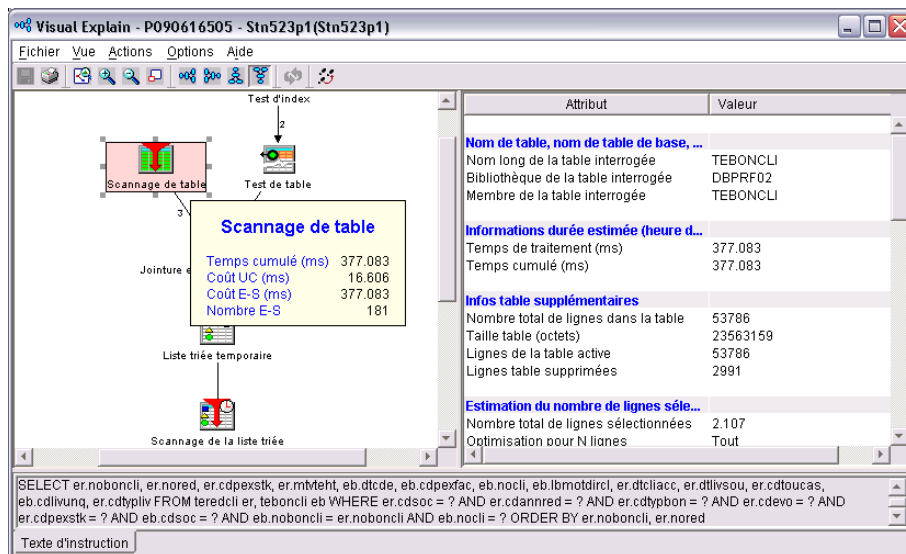
Récapitulatif
Instructions

Fermeture Aide ?

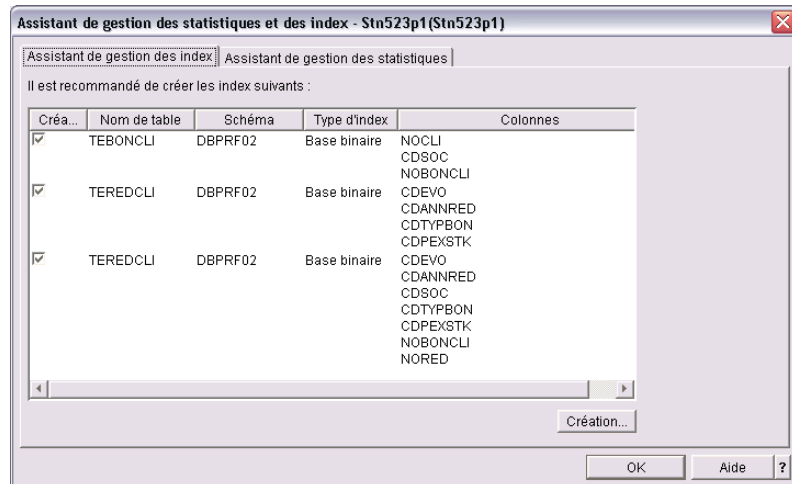
Cache de plan



Visual Explain



Index Advisor



Les pièges du « chronomètre »

- Données « froides » versus données « chaudes »
- Régime de croisière atteint après 2 exécutions en SQL dynamique
- Régime de croisière atteint après 1 exécution en SQL statique
- SQE peut mettre de coté le résultat d'une requête et ne pas exécuter à nouveau votre requête
- La clause OPTIMIZE for N/all rows

Pour vous aider

2 approches :

- Prestation sur site (2 jours)
 - 'Checkup de votre IBM i' avec votre équipe
 - Fourniture d'un rapport de préconisations
 - Contre-visite (optionnelle) après mise en oeuvre
- Cours
 - 'Analyse et Améliorations de Performance SQL sous DB2 for i' (OL40FR, prochaine session en France : 24-27 mai 2011)

Merci pour
votre présence
et
votre participation