

---

## Analyse de performance sous Oracle

---

### Outils d'analyse de performance

Une des forces principales d'Oracle est la disponibilité de plusieurs outils permettant d'analyser l'état et la performance de la BD. Un des principaux outils est un ensemble de vues (VIEW) dont les noms ont le préfixe commun V\$. Par exemple, les vues V\$LOCK et V\$LOCKED\_OBJECTS contiennent de l'information sur les ressources qui sont verrouillées dans la BD. Pour l'analyse de performance, il existe une vue du nom de V\$SQL\_PLAN\_STATISTICS\_ALL contenant des statistiques sur la mémoire et le temps requis pour exécuter une requête selon le plan d'exécution choisi par Oracle. Parmi les colonnes les plus importantes de cette vue, on retrouve les suivantes :

OPERATION	: Le nom de l'opération du plan d'exécution (ex : NESTED LOOPS, HASH JOIN).
COST	: Le coût de l'opération tel qu'estimé par l'optimiseur de requête. En mode d'optimisation par coût, cette valeur sert de justification au plan d'exécution choisi par Oracle.
LAST_CR_BUFFER_GETS	: Le nombre de blocs lus de la cache pour chaque opération, lors de la dernière exécution de la requête.
LAST_ELAPSED_TIME	: Le temps (en microsecondes) passé dans chaque opération, lors de la dernière exécution de la requête.
LAST_STARTS	: Le nombre d'itérations de chaque opération, lors de la dernière exécution de la requête.

Étant donné que le plan d'exécution est sous la forme d'un arbre (où chaque nœud correspond à une opération), le plan d'exécution peut être difficile à visualiser dans la table V\$SQL\_PLAN\_STATISTICS\_ALL. Ainsi, la commande SET AUTOTRACE ON d'Oracle permet de simplifier cette tâche en affichant, après l'exécution de chaque requête, le plan d'exécution utilisé par Oracle. En plus du plan d'exécution, l'activation du mode AUTOTRACE génère également l'affiche de statistiques globales de l'exécution :

**recursive calls:** Le nombre de requêtes SQL (appels récursifs) générées par Oracle pour mettre à jour les tables internes lors de l'exécution.

**db block gets:** Le nombre de blocs lus en mode d'isolation<sup>1</sup> CURRENT.

**consistent gets:** Le nombre de blocs lus en mode d'isolation CONSISTENT.

**physical reads:** Le nombre total de blocs lus à partir du disque.

**redo size:** Le nombre d'octets générés pour le journal permettant de refaire l'opération (redo log).

**bytes sent:** Le nombre d'octets envoyés de la BD au client par le réseau.

**bytes received:** Le nombre d'octets envoyés du client à la BD par le réseau.

**round-trips to/from:** Le nombre total de messages envoyés par et reçu du client à travers le réseau.

**sorts (memory):** Le nombre d'opérations de tri effectuées entièrement en mémoire centrale.

**sorts (disk):** Le nombre d'opérations de tri ayant nécessité au moins une écriture au disque.

**rows processed:** Le nombre de lignes manipulées durant l'exécution.

## Analyse de performance sous SQLDeveloper

L'outil SQLDeveloper d'Oracle permet de visualiser de manière plus conviviale le plan et les statistiques d'exécution du mode AUTOTRACE. Pour activer ce mode, il faut cliquer sur l'icône associé, tel que montré à la Figure 1. À noter que des colonnes de statistiques (ex : LAST\_ELAPSED\_TIME) peuvent être rajoutées/enlevées à partir du menu des préférences de SQLDeveloper.

## Références

- Optimisation de requêtes SQL dans Oracle : [http://download.oracle.com/docs/cd/B19306\\_01/server.102/b14357/ch8.htm](http://download.oracle.com/docs/cd/B19306_01/server.102/b14357/ch8.htm)
- Les vues de performance dynamique dans Oracle : [http://download.oracle.com/docs/cd/B10501\\_01/server.920/a96533/sqlviews.htm#40886](http://download.oracle.com/docs/cd/B10501_01/server.920/a96533/sqlviews.htm#40886)

---

1. Le mode d'isolation est relié à la gestion de l'intégrité des données dans un contexte d'accès concurrents.

Icône d'activation du mode AUTOTRACE

Page de démarrage x uqac ora8trd157\_030 x

43,71099854 secondes

Feuille de calcul Query Builder

```
select count(*) from t1, t2 where T1.col1 = t2.col1
```

Résultat de requête x Enregistrer la trace automatiquement x

SQL | 43,711 secondes

OPERATION	OBJECT_NAME	COST	LAST_CR_BUFFER_GETS	LAST_ELAPSED_TIME
SELECT STATEMENT		4412		
SORT (AGGREGATE)			4800	32307298
HASH JOIN		4412	4800	26712546
Access Predicates T1.COL1=T2.COL1				
TABLE ACCESS (FULL)	T1	759	2815	112213
TABLE ACCESS (FULL)	T2	555	1985	136900

V\$STATNAME Name	V\$MYSTAT Value
Batched IO (bound) vector count	0
Batched IO (full) vector count	0
Batched IO (space) vector count	0
Batched IO block miss count	0
Batched IO buffer defrag count	0
Batched IO double miss count	0
Batched IO same unit count	0
Batched IO single block count	0
Batched IO slow jump count	0
Batched IO vector block count	0
Batched IO vector read count	0
Batched IO zero block count	0
Block Cleanout Optim referenced	0
CCursor + sql area evicted	0
CPU used by this session	3226
CPU used when call started	3226
CR blocks created	0
Cached Commit SCN referenced	0
Clusterwide global transactions	0
Clusterwide global transactions spanning RAC nodes	0
Commit SCN cached	0

FIGURE 1 – Visualisation du plan et des statistiques d'exécution dans SQLDeveloper