TP2 Architecture Oracle (3h)

1. Préalable

L'optimisation est entendue comme étant la capacité à adapter l'architecture d'un système aux différents besoins des applications clientes (in fine, l'objectif est de satisfaire les usagers). Il va donc s'agir d'un ensemble d'ajustements au niveau de l'instance et de la base de données qui va permettre d'atteindre des objectifs de **performance** (c'est à dire augmenter le débit transactionnel, et réduire le temps des réponses).

2. Serveur Oracle

Un serveur Oracle est, en grande majorité, constitué d'une instance (structures cache + processus) et d'une base de données (ensemble de fichiers).
Quelques questions d'ordre général, et qui concerne le serveur, vous sont posées.

1. consulter la vue (v$version) portant sur la version du SGBD Oracle sous-jacent
2. consulter l'attribut server de la vue v$session pour connaître l'architecture client-serveur retenue pour servir les connexions utilisateurs (architecture dédiée ou partagée).
3. consulter la vue (v$option) portant sur les fonctionnalités du serveur de données et répondre à des interrogations telles que : de quelles options disposons-nous ?

3. L'instance

Une vue dynamique donne également quelques renseignements à propos de l'instance :

1. consulter la vue portant sur l'instance (v$instance) et répondre à des interrogations telles que : quel est le nom de l'hôte sur lequel tourne l'instance, et depuis quand l'instance est démarrée,

3.1 Structures mémoire

3.1.1 Taille des sous-structures de la SGA

Vous consulterez les vues portant sur la structure mémoire System Global Area, ou SGA (v$sga ou show SGA, v$sgainfo). La SGA est divisée en différentes sous-structures, qui chacune possède un rôle clé, dans l'efficacité et la sécurité des accès aux données. Ainsi la mémoire partagée (shared pool) correspond au cache des dernières requêtes (library cache) et du dictionnaire de données (dictionary cache). Le cache de données (data buffer cache) est la zone de transit et d'écriture pour ce qui concerne les blocs de données. Le tampon de journalisation (redo log buffer) conserve l'information sur les dernières transactions en cours.
Vous répondrez à des interrogations telles que :

1. quelle est la taille (en Mo) allouée à la mémoire partagée (shared pool),
2. quelle est la taille (en Mo) allouée au tampon de données (data buffer cache),
3. quelle est la taille (en Mo) allouée au tampon de journalisation (redo log buffer),
4. quelle est la taille totale (en Go) allouée à la SGA

Pour aller dans plus de détail, vous consulterez éventuellement la vue v$sgastat.

3.2 Mémoire "LibraryCache"

3.2.1 Indicateur de performance concernant la mémoire partagée

La mémoire partagée contient la "LibraryCache" et le "DictionaryCache". Nous évaluons ici la performance de la "LibraryCache". La vue système v$librarycache donne des informations sur les derniers ordres (SQL, Trigger et Procédure PL/SQL, ...) exécutés. L'objectif de l'indicateur à construire est d'évaluer le nombre d'exécutions qui n'ont pas nécessité de charger l'ordre en mémoire (pins) au regard du nombre de rechargements (reloads). Vous interrogerez v$librarycache pour construire le "library cache hit ratio" en question (somme de ("pins" - reloads) / somme de "pins"). Vous vérifierez que sa valeur est très proche de 1. Que pouvez vous en conclure si cette valeur est proche de 1 ? Que faire si elle est très inférieure à 1 ?

3.2.2 Les derniers ordres du cache de requêtes

Tous les ordres SQL (mais pas que) transitent par la mémoire "LibraryCache". Vous exploiterez les vues système afférentes à cette zone mémoire : v$sqlarea, v$sql, v$sqltext et v$sql_plan pour tracer et évaluer les dernières requêtes traitées par le système. Différents exemples de requête (allant dans ce sens) sont donnés. Vous construirez une première procédure nommée User_Activity permettant d'afficher des informations sur les accès aux données d'un usager dont le schéma utilisateur est passé en paramètre d'entrée. Vous construirez une seconde procédure Costly_Cursors qui permet de renvoyer les 10 requêtes (appelées cursors dans le contexte de la "LibraryCache") les plus coûteuses tous usagers confondus. Le coût est fonction du nombre de blocs accédés sur disque, du temps de calcul au niveau du processeur (cpu_time), du temps de traitement total (elapsed_time). Attention dans v$sqlarea, les données sont cumulées au fur et mesure que le même ordre est exécuté (le nombre d'exécutions est donné pat l'attribut executions). Nous n'allons pas lister l'ensemble des attributs de ces vues systèmes, mais seulement les attributs exploités dans les exemples :

1. SQL_ID : identifiant du curseur
2. SQL_TEXT : ordre SQL / PL/SQL
3. DISK_READS : nombre de blocs en accès disque
4. PARSING_SCHEMA_NAME : schéma utilisateur
5. CPU_TIME : Temps CPU (en microsecondes) utilisé pour le curseur pour le cycle parse/execute/fetch/
6. ELAPSED_TIME : Temps écoulé (en microsecondes) utilisé pour le curseur pour le cycle parse/execute/fetch/
7. EXECUTIONS : nombre de fois où l'ordre a été lancé
8. BUFFER_GETS : nombre de blocs exploités depuis le cache de données

```
set linesize 200
col req for a80
```

```
select sql_id, substr(sql_text,1,80) as req, disk_reads from v$sqlarea where
    parsing_schema_name =user;

col parsing_schema_name for a16
select parsing_schema_name, sql_id, substr(sql_text,1,80) as req from v$sqlarea order by
    first_load_time ;

select parsing_schema_name, sql_id, substr(sql_text,1,80) as req from v$sqlarea where
    parsing_schema_name<>'SYS';


select to_char(logon_time, 'DD/MM/YYYY HH24:MI:SS') , username, program, sql_text
from v$session , v$sqlarea
where v$session.sql_address = v$sqlarea.address
order by username, program;


select r.sql_id, disk_reads, elapsed_time, username from v$sql r, v$session s where
    s.sql_id = r.sql_id and type='USER';

select sql_FullText,(cpu_time/100000) "Cpu Time (s)"
, (elapsed_time/1000000) "Elapsed time (s)"
, fetches, buffer_gets, disk_reads, executions
FROM v$sqlarea WHERE Parsing_Schema_Name ='P00000009432'
AND rownum <50
order by 3 desc;
```

Listing 1 – exemples

3.3 Tampon de données "data buffer cache"

3.3.1 Indicateur de performance d'accès

Un indicateur nommé buffer_hit_ratio permet d'évaluer la performance en matière d'accès aux données. Le calcul se fait de la manière suivante :

```
Select 1- (phy.value / ( cons.value + db.value - phy.value))
from v$sysstat phy, v$sysstat cons, v$sysstat db
where phy.name ='physical reads' and cons.name ='consistent gets' and db.name ='db block
    gets';
```

Listing 2 – processus

Vous chercherez à expliquer en quoi il consiste.

3.3.2 En savoir plus sur les blocs du cache

Un exemple d'exploitation de la vue v$bh (jointure avec dba_objects) est donné.

```
select file#, block#, class#, dirty, objd, object_name, owner
from v$bh, dba_objects where dirty = 'Y' and objd = object_id;
```

Listing 3 – exemples

Vous expliquerez le résultat obtenu.
Le numéro de classe est un indicateur du type de bloc. Quelques valeurs possibles et leurs significations sont listées ici : (1 = 'data block'), (2 = 'sort block'), (3, 'save undo block'), (4, 'segment header'), (5,

'save undo header'), (6, 'free list'), (7, 'extent map'), (18, 'undo block') ... . Comment lister tous vos blocs de données qui se situent en mémoire cache ? Comment connaître le nom des objets associés à ces blocs ? Comment identifier l'usager qui est le plus gros consommateur du cache de données ?

3.4 Processus

Une requête vous est donnée pour identifier les processus d'arrière-plan interagissant avec les structures mémoire et les fichiers de données. Cette requête exploite deux vues systèmes associées aux processus Oracle. La vue v$bgprocess (bg pour background) contient les informations requises pour caractériser les processus d'arrière-plan. Vous chercherez à retrouver les processus d'arrière-plan décrits en cours. Que retrouve t'on comme processus dans v$process qui ne sont pas listés dans v$bgprocess ?

```
select p.pid, bg.name, bg.description, p.program
from v$bgprocess bg, v$process p
where bg.paddr = p.addr order by p.pid;
```

Listing 4 – processus

3.5 Lien entre les structures logiques et physiques

Vous traiterez les questions suivantes :

1. consulter les tablespaces définis (dba_tablespaces)
2. consulter l'emplacement des fichiers de données (v$datafile) ;
3. consulter l'emplacement des fichiers journaux (v$logfile) ;
4. consulter l'emplacement des fichiers de contrôle (v$controlfile) ;
5. faire le lien entre espace de table et fichier de données au travers de la requête : `select tablespace_name, file_name from dba_data_files ;`. Combien de fichiers sont asservis à chaque tablespace ? ;