SUPPORT DE COURS ORACLE

CHAPITRE 6

LES SEGMENTS D'ANNULATION (ROLLBACK SEGMENTS) COMPLEMENTS

Les segments d'annulation (compléments)

Information sur les segments d'annulation

La vue dba_rollback_segs fournit des informations générales sur les segments d'annulation.

La vue v$rollname donne les noms des segments d'annulation.

La vue v$rollstat fournit des statistiques sur les segments d'annulation (RBS) Suivent la description de quelques colonnes :

Nom de la colonne	Description
USN	Numéro du RBS
EXTENTS	Nombre d'extents dans le RBS
RSSIZE	Taille du RBS en octets
WRITES	Nombre d'octets écrits dans le RBS
XACTS	Nombre de transactions actives
GETS	Nombre d'en-têtes de RBS demandés
WAITS	Nombre d'en-têtes de RBS demandés qui ont entraînés des attentes
OPTSIZE	Valeur du paramètre OPTIMAL pour le RBS
HWMSIZE	Taille maximale atteinte par le RBS

Gestion des segments d'annulation

Oracle ne permet pas l'utilisation du paramètre PCTINCREASE de la clause STORAGE pour les segments d'annulation.

Le paramètre OPTIMAL (propre aux segments d'annulation) de la clause STORAGE laisse le segment d'annulation s'étendre puis une fois les extents libérés, par une ou plusieurs transactions, le segment d'annulation retrouve l'espace spécifié par le paramètre OPTIMAL. Cette désallocation d'espace se produit au moment où une nouvelle transaction accède au segment d'annulation.

Exemple :

alter rollback segment mon_rbs storage (optimal 500 k);

Il faut noter que, si le paramètre OPTIMAL est affecté d'une valeur faible, l'allocation puis la désallocation d'extents peuvent réduire les performances et favoriser la survenue de l'erreur *ORA-01555*.

Il est alors possible de choisir de réduire manuellement l'espace d'un segment d'annulation en lui spécifiant sa taille ou en le ramenant à sa taille donnée par OPTIMAL à l'aide de la commande :

alter rollback segment mon_rbs shrink to 800k

alter rollback segment mon_rbs shrink

Volume d'activité

Afin de connaître l'activité de chaque segment d'annulation on pourra effectuer une requête utilisant les deux vues suivantes :

```
select n.name "Segment d'annulation", s.writes "Oct.écrits"
from v$rollname n, v$rollstat s
where n.usn=s.usn;
```

Segment d'annulation	Oct.écrits
SYSTEM	4930
RB1	486
RB2	272
RB3	54
RB4	54
RB5	54
RB6	54
RB7	54
MON_RBS2	5942

Utilisateurs et segments d'annulation

La requête suivante permet de connaître l'utilisation des segments d'annulation réalisée par chaque utilisateur :

```
select    r.name "Segment d'annulation",
          s.username "Utilisateur Oracle",
          s.osuser "Utilisateur OS",
          s.terminal
from      v$session s,
          v$transaction t,
          v$rollname r
where     s.saddr=t.ses_addr(+)
  and     r.usn=t.xidusn
```

Segment d'annulation	Utilisateur Oracle	Utilisateur OS	TERMINAL
MON_RBS2	SYSTEM	Administrateur	BOTERO
MON_RBS2	SYSTEM	Administrateur	BOTERO
RB1	JMI	Jmi	CAILLEBOTTE
RB3	SCOTT	Administrateur	BOTERO

4 lignes sélectionnées

Détermination du nombre de segments d'annulation

Afin de déterminer le nombre de segments d'annulation nécessaires à une base, Oracle recommande d'utiliser la *règle des quatre*, qui consiste à diviser par 4 le nombre maximum de transactions concurrentes pour l'instance. L'entier supérieur, résultant de cette division, donnera le nombre de segments d'annulation, avec un minimum de 8 et un maximum de 50.

Dimensionnement des segments d'annulation

1. La colonne used_ublk de la vue v$transaction retourne le nombre de blocs de données utilisés par chaque transaction pour les informations d'annulation. Afin d'avoir une idée du volume maximal d'informations d'annulation généré par les transactions, la requête suivante peut être lancée à intervalles réguliers quand l'instance est fortement sollicitée :

```
select      max(used_ublk) volume_max
from        v$transaction
```

2. L'arrondi supérieur de la valeur obtenue permet de déterminer la taille des extents *initial* et *next*

3. Chaque segment d'annulation a besoin au minimum de 2 extents (valeur minimale de *minextents* pour les segments d'annulation). Des tests effectués dans les laboratoires d'Oracle ont montré qu'en fixant à 20 la valeur de *minextents*, la probabilité de voir apparaître une erreur *ORA-01555* était réduite. Pour configurer ainsi les segments d'annulation, le DBA doit alors disposer d'espace disque suffisant

4. Pour réduire encore la probabilité de voir apparaître l'erreur *ORA-01555*, il est préférable de choisir un autre moment pour exécuter les requêtes longues, lorsque, par exemple, l'activité DML est plus faible, ou d'affecter un segment d'annulation spécifique à un requête particulière.

Exercice sur les Rollback Segments :

Objectif : Gestion des Rollback segments et des transactions.

1. Visualiser tous les segments d'annulation de l'instance.
2. Affecter à un des segments d'annulation une valeur OPTIMAL. (légèrement supérieure à la somme des extents minimum le constituant)
3. Sous d'autres connexions, ouvrir 2 à 3 nouvelles transactions effectuant des opérations DML. Une de ces transactions pourra éventuellement être affectée à un segment d'annulation nommé à l'ouverture de la transaction.
4. Visualiser toutes les transactions en cours, et pour chaque transaction le nom de l'utilisateur Oracle, ainsi que le nom de l'utilisateur système.
5. Pour chaque segment d'annulation, visualiser son nom, le nombre d'extents dont il est constitué, le nombre de transactions qui y accèdent, sa taille optimale, sa taille réelle, sa taille maximale.
6. Forcer le segment d'annulation de la question 2, à sa valeur optimale. Est-ce possible et pourquoi ?
7. Forcer un des segments d'annulation à une taille que vous aurez choisie.

SUPPORT DE COURS ORACLE

CHAPITRE 7

ARCHITECTURE ET MISE EN ŒUVRE D'ORACLE

Architecture et mise en œuvre d'Oracle

- Une instance Oracle est la combinaison :
  - de structures mémoires
  - de certains process Oracle (les process détachés)
- Une instance démarrée peut monter la base de données constituée:
  - de fichiers

Structure de la mémoire

- Trois types de zones:
  - zones réservées au code l'applicatif
  - zone globale système (la SGA)
  - zone globale programme (la PGA)

Zones réservées au code de l'applicatif

- Composées de parties de la mémoire permettant de stocker le code des programmes en cours d'exécution:
  - noyau (zone séparée des autres)
  - outils Oracle (SQL*Plus, SQL*Forms etc...)
  - applicatif faisant appel à Oracle
- Accessibles uniquement en lecture, elles peuvent être partagées ou exclusives (sauf certains OS)

La SGA: System Global Area

- Groupe de structures de mémoire **partagée** contenant des données et des informations de contrôle (aussi appelée Shared Global Area)
- Allouée au démarrage du serveur
- En mémoire non-paginée et non-swappée
- Taille déterminée par différents paramètres:
  - DB_BLOCK_SIZE (taille d'un bloc)
  - DB_BLOCK_BUFFERS (nombre de tampons)
  - LOG_BUFFER (taille de la zone tampon de reprise: redo log buffer)
  - SHARED_POOL_SIZE (taille de la shared pool)

Zones de la SGA: la Shared Pool

- la zone Pool Partagé (la Shared Pool) contient:
  - des zones de requêtes SQL partagées (textes des ordres SQL, formes analysées, plan d'exécution) ou privées (serveur multi-thread) gérées par un algorithme LRU (Least Recent Used)
  - un cache mémoire du dictionnaire de données
- sa taille est déterminée par le paramètre: SHARED_POOL_SIZE
- pour libérer manuellement la Shared Pool utiliser:
  - ALTER SYSTEM FLUSH SHARED POOL.

Zones de la SGA: le Database Buffer Cache

- le cache des buffers de la base de données est un ensemble de copies des blocs de données lus à partir des fichiers
- son espace est géré à l'aide de deux listes:
  - une liste de tampons modifiés mais non encore écrits sur disque (*dirty list*)
  - une liste de tampons les moins utilisés récemment (*least recently used list* ou LRU) qui contient les tampons libres (pouvant être utilisés), les tampons en cours d'utilisation par certains processus, et les tampons modifiés mais non encore transférés dans la 1ère liste (cf. paramètre DB_BLOCK_MAX_SCAN et processus DBWR)

Zones de la SGA: le Database Buffer Cache

- Définitions:
  - le cache *miss*: le bloc de données n'est pas présent en mémoire; il est lu à partir du disque
  - le cache *hit* : le bloc de données est présent en mémoire; l'accès est beaucoup plus rapide
- Contenu de la liste LRU:

Types de buffers	Description
Free buffers	Buffers qui n'ont pas été modifiés et sont disponbles
Pinned buffers	Buffers qui sont en cours d'accès
Dirty Buffers	Buffers modifiés qui doivent écrits sur disque

Zones de la SGA: le Database Buffer Cache

- Définitions:

  Le **cache hit-ratio** (taux de réussite du cache ou ratio de présence) permet de détecter si la taille du cache des tampons de données est correcte. 1 serait une valeur idéale. Ce ratio est calculé de la manière suivante:

  (lectures logiques – lectures physiques) / lectures logiques

Zones de la SGA: la Large Pool

- C'est une zone de mémoire optionnelle
- Elle est utilisée quand Oracle demande de grandes zones contiguës de mémoire à l'intérieur du pool partagé
- Pour créer la Large Pool, il faut affecter une valeur au paramètre LARGE_POOL_SIZE

Zones de la SGA: le Redo log buffer

- la zone de tampon de reprise est un tampon circulaire qui contient des informations relatives aux modifications apportées à la base de données; ces informations peuvent être utilisés en cas de reprise.
- le contenu de ce tampon est écrit dans le(s) fichier(s) de reprise (Redo log file) par la processus LGWR.

Zones de la SGA: autres informations

- les autres informations que peut contenir la SGA sont:
  - informations communiquées entre processus telles que des informations relatives au verrouillage
  - informations relatives au dictionnaire

La PGA: Global Program Area

- la zone programme globale contient des données et informations relatives à un processus (serveur ou d'arrière plan).
- une zone PGA est allouée par Oracle lorsqu'une session est ouverte.
- la PGA est modifiable et non partagée
- le contenu de la PGA varie selon que l'on utilise un serveur multi-threaded ou non

Les processus Oracle

- Deux types de processus SGBD:
  - les processus d'arrière-plan ou processus détachés (Background process) effectuent d'une façon asynchrone l'ensemble des tâches du SGBD pour tous les utilisateurs.
  - les processus serveurs permettent d'associer à un ensemble de processus utilisateurs un seul processus serveur et de minimiser ainsi la charge de l'OS.
- Les processus utilisateurs:
  - un processus utilisateur est créé lorsqu'un outil Oracle est exécuté par un utilisateur ou à travers une application

Les Background process

- DBWR (Database Writer) obligatoire
- LGWR (Log Writer) obligatoire
- CKPT (Checkpoint)
- SMON (System monitor) obligatoire
- PMON (Process monitor) obligatoire
- ARCH (Archiver)
- RECO (Recoverer) option supplémentaire
- LCKn (Lock) option supplémentaire
- Dnnn (Dispatcher)
- Listener
- Snnn (Shared Server)
- Parallel Query (Pnnn) option supplémentaire
- SNPn (Snapshot Refresh) option supplémentaire

PMON

- PMON
  - nettoie les connexions terminées de facon anormale
  - défait les transactions non validées
  - libère les verrous qui avaient été posés par un process qui s'est terminée en erreur
  - redémarre les serveurs partagés et les process *dispatcher* en erreur

SMON

- SMON
  - réalise la restauration automatique d'instance
  - récupère l'espace occupé par des segments temporaires qui ne sont plus utilisés
  - fusionne les zones contiguës d'espace libre dans les fichiers de données

DBWR

- DBWR
  - Gére le Database Buffer Cache
  - Ecrit périodiquement les modifications dans les fichiers de données mais diffère les écritures en vue d'optimiser les E/S
- DBWR écrit les buffers dirty sur le disque quand:
  - la dirty list atteint une valeur limite
  - un process a parcouru un nombre spécifié de buffers dans la LRU list sans trouver un buffer libre
  - un time-out se produit ou un checkpoint de DBWR se produit

LGWR

- LGWR
  - Assure l'écriture des Redo log buffers dans le fichier de reprise (Redo log)
  - Si il existe des groupes de fichiers Redo log, LGWR effectue une écriture synchrone sur tous les fichiers
  - le processus est activé:
    - lorsqu'un commit se produit
    - lorsque le tiers de la zone tampon de reprise est plein
    - lorsque le DBWR effectue une écriture des tampons modifiés sur disque

CKPT

- Pendant un point de synchronisation (*checkpoint*), le DBWR écrit les données modifiées du cache base de données sur disque. Un numéro de séquence de checkpoint est mis à jour dans chaque fichier de la base et dans le(s) fichier(s) de contrôle.
- Pour améliorer les performances il est possible d'activer CKPT qui prend en charge l'opération mené par LGWR

ARCH

- Copie les fichiers redo log actifs sur un support de stockage déterminé (disque ou bande), chaque fois que le LGWR passe sur un nouveau groupe.
- N'est actif que si les fichiers de reprise sont utilisés en mode ARCHIVELOG

Dnnn

- Permet le partage d'un nombre limité de processus serveurs par les processus utilisateurs
- Le nombre de processus Dnnn (Dispatchers) est fixé par l'administrateur.
- Ces processus Dispatchers sont utilisés en configuration '*multi-threaded server*'

Listener

- Se met en attente des connexions des utilisateurs et les aiguille vers un processus Dispatcher.
- Ce processus Listener est utilisé en configuration '*multi-threaded server*'

RECO, LCKn, Parallel Query, SNPn

- RECO (Recoverer) résout les erreurs concernant une transaction distribuée
- LCKn (Lock) réalise le verrouillage inter-instance dans un système *parrallel server*
- Parallel Query (Pnnn) permet le parallélisme des requêtes
- SNPn (Snapshot Refresh) réalise les rafraîchissements automatiques des *snapshots* (tables répliquées en lecture seule)

Les processus serveurs

Le rôle d'un processus serveur est de traiter les requêtes soumises par un (ou des) process utilisateur.

Le processus serveur permettra ainsi de:

- analyser et exécuter les commandes SQL
- rechercher et modifier les données en SGA
- transférer les blocs de données nécessaires des fichiers sur disque vers la SGA
- communiquer les résultats de requêtes aux applications
- maintenir en mémoire les blocs les plus récemment accédés en suivant l'algorithme LRU

Les architectures multiprocessus

- Architecture combinée:
  - pour chaque utilisateur, le processus utilisateur et le processus serveur sont combinés en un seul processus utilisateur; il n'y donc pas de processus serveur;
  - il faut que les processus de l'OS permette la séparation entre le code de l'applicatif et celui d'Oracle; c'est le cas de VMS et non d'Unix.

Les architectures multiprocessus

- Architecture à serveur dédié:
  - à chaque processus utilisateur correspond un processus serveur, dit processus serveur dédié. Les 2 processus sont séparés. Le processus utilisateur est généralement sur une station de travail, le processus serveur dédié sur la machine servant de serveur de données

Les architectures multiprocessus

- Architecture à serveur partagé (*multi-threaded server)*:
  - ici plusieurs processus utilisateurs peuvent partager un même processus serveur.

Gestion et surveillance de la base

- Outils de connexion:
  - server manager (à partir d'Oracle7.2 jusqu'à Oracle 8i)
  - sqlplus à partir d'Oracle 8i

Les états d'une instance

- SHUTDOWN: l'instance est arrêtée
- NOMOUNT: l'instance démarrée, permet de créer la base; la zone mémoire et les processus ne sont encore associés à aucune base de données.
- MOUNT: le fichier de contrôle est ouvert pour cette instance qui est placée dans un état de maintenance; l'administrateur peut administrer la base.
- OPEN: tous les fichiers définis dans le fichier de contrôle sont ouverts.

SUPPORT DE COURS ORACLE

CHAPITRE 8

NLS : NATIONAL LANGUAGE SUPPORT

NLS: National Language Support

- Un jeu de caractères ou *character set* définit une table de correspondance entre des caractères et des codes binaires.

Jeu de caractères pour une base Oracle

- Le jeu de caractères (NLS_CHARACTER) utilisé par la base est figé lors de la création de la base.
- Ce jeu de caractères effectue la correspondance entre binaire et caractère stocké.
- La vue SYS.PROPS$ permet la consultation du character set

La vue SYS.PROPS$

```
SQL> select * from SYS.PROPS$;

NAME                           VALUE$     COMMENT$
------------------------------ ---------- ------------------------------
DICT.BASE                      2          dictionary base tables version #
NLS_LANGUAGE                              AMERICAN  Language
NLS_TERRITORY                             AMERICA   Territory
NLS_CURRENCY                              $         Local currency
NLS_ISO_CURRENCY                          AMERICA   ISO currency
NLS_NUMERIC_CHARACTERS                    .,        Numeric characters
NLS_CALENDAR                              GREGORIAN Calendar system
NLS_DATE_FORMAT                           DD-MON-YY Date format
NLS_DATE_LANGUAGE                         AMERICAN  Date language
NLS_CHARACTERSET                          WE8ISO8859P1 Character set
NLS_SORT                       BINARY     Linguistic definition
NLS_NCHAR_CHARACTERSET                    WE8ISO8859P1 NCHAR Character set
NLS_RDBMS_VERSION                         8.0.5.0.0 RDBMS version for NLS parameters
GLOBAL_DB_NAME                            ORACLE.WORLD Global database name
EXPORT_VIEWS_VERSION                      7         Export views revision #

15 ligne(s) sélectionnée(s).
```


Jeu de caractères pour un client

- Le jeu de caractères utilisé par un ordinateur client doit être identique à celui utilisé par le client oracle.
- Le jeu de caractères est fixé sur le poste client par la variable NLS_LANG.

  Elle est au format:
  LANGUAGE_TERRITORY.CHARACTERSET

  Exemple: NLS_LANG FRENCH_FRANCE.WE8ISO8859P1

  Cette variable est une variable d'environnement sous Unix, ou fixée dans la base de registre sous Windows 95/98/NT.

La variable NLS_LANG

- La partie LANGUAGE de NLS_LANG agit sur la langue des messages utilisés par les outils clients Oracle, sur le nom des jours et des mois ainsi que le format de présentation si celui-ci n'est pas modifié.
- La partie TERRITORY de NLS_LANG agit sur le nom des jours et des mois, ainsi que la numérotation des jours et des semaines, sur le format des dates, le symbole monétaire…

Gestion des paramètres NLS au niveau de la base

- Les paramètres NLS peuvent être spécifiés dans le fichier *init<sid>.ora*. Ils affectent alors la base de données et non le poste client.

Les paramètres NLS par défaut de la base

- Valeurs par défaut utilisées lors de la création de la base de données:

```
SQL> select * from NLS_DATABASE_PARAMETERS;
PARAMETER                      VALUE
------------------------------ ----------------------------------------
NLS_LANGUAGE                   AMERICAN
NLS_TERRITORY                  AMERICA
NLS_CURRENCY                   $
NLS_ISO_CURRENCY               AMERICA
NLS_NUMERIC_CHARACTERS         .,
NLS_CALENDAR                   GREGORIAN
NLS_DATE_FORMAT                DD-MON-YY
NLS_DATE_LANGUAGE              AMERICAN
NLS_CHARACTERSET               WE8ISO8859P1
NLS_SORT                       BINARY
NLS_NCHAR_CHARACTERSET         WE8ISO8859P1
NLS_RDBMS_VERSION              8.0.5.0.0
```


Les paramètres NLS spécifique à une instance

- Les paramètres NLS peuvent être spécifiés dans le fichier *init<sid>.ora*. Ils affectent alors l'instance de la base de données et non le poste client.

```
SQL> select * from NLS_INSTANCE_PARAMETERS;
PARAMETER                      VALUE
------------------------------ ----------------------------------------
NLS_LANGUAGE                   AMERICAN
NLS_TERRITORY                  AMERICA
NLS_SORT
NLS_DATE_LANGUAGE
NLS_DATE_FORMAT
NLS_CURRENCY
NLS_NUMERIC_CHARACTERS
NLS_ISO_CURRENCY
NLS_CALENDAR
```

Les paramètres NLS d'une session

- Paramètres utilisés par la variable NLS_LANG du poste client:

```
SQL> select * from NLS_SESSION_PARAMETERS;
PARAMETER                      VALUE
------------------------------ ----------------------------------------
NLS_LANGUAGE                   FRENCH
NLS_TERRITORY                  FRANCE
NLS_CURRENCY                   F
NLS_ISO_CURRENCY               FRANCE
NLS_NUMERIC_CHARACTERS         ,.
NLS_CALENDAR                   GREGORIAN
NLS_DATE_FORMAT                DD/MM/YY
NLS_DATE_LANGUAGE              FRENCH
NLS_SORT                       FRENCH
```


- Valeurs actuelles des variables NLS
- Select * from V$NLS_PARAMETERS;

- Valeurs pouvant être utilisées
- Select * from V$NLS_VALID_VALUES;

SUPPORT DE COURS ORACLE

CHAPITRE 9

CREATION D'UNE BASE DE DONNEES

L' administrateur de bases Oracle et les droits systèmes

Un administrateur de bases Oracle doit avoir des droits systèmes particuliers :

- *Sur Unix*

  l' administrateur de bases Oracle doit avoir un compte Unix particulier, généralement appelé Oracle, qui appartient à un groupe prédéfini Unix obligatoirement dénommé DBA. Les fichiers composant le logiciel et les process d'Oracle, appartiendront à cet utilisateur. Pour certaines étapes de l'installation (notamment création du compte Oracle, mise à jour des fichiers de démarrage, et des paramètres système) l'administrateur de bases Oracle devra posséder les privilèges 'root'.

- *Sur NT*

  le DBA devra appartenir au groupe administrateur, notamment pour démarrer les services Oracle sur le serveur.

Étapes de création d'une base Oracle

- Calibrer vos besoins

Avec un éditeur de texte:

- Créer le fichier INIT*SID*.ORA

En lançant Server Manager jusqu'à la version Oracle 8.0 ou sous SQL*PLUS à partir de la version Oracle 8i :

- Démarrer l'instance sans monter la base
- Lancer l'exécution de l'ordre CREATE DATABASE

A l'aide de requêtes SQL ou d'outils visuels (Entreprise Manager jusqu'en 8.0, ou DBA Studio à partir de la 8i)

- Créer les tablespaces nécessaires à la base et aux applicatifs
- Créer les rollback segments calibrés pour les applicatifs
- Créer les utilisateurs et les droits associés
- Lancer les scripts de création d'objets de la base (tables, clusters, index, vues, séquences, etc.)

Création physique d'une base

L'exécution de l'ordre CREATE DATABASE :

- réalise la création des fichiers datafiles, control files, logfiles de la base
- réalise la création du tablespace SYSTEM et du rollback segment SYSTEM
- réalise la création du dictionnaire de données
- réalise la création des utilisateurs SYS et SYSTEM
- spécifie le jeu de caractères internes
- monte et ouvre la base de données

Création physique d'une base: exemple

```
CREATE DATABASE newtest
CONTROLFILE REUSE
LOGFILE GROUP 1 ('diskb:log1.log', 'diskc:log1.log') SIZE 50K,
    GROUP 2 ('diskb:log2.log', 'diskc:log2.log') SIZE 50K
MAXLOGFILES 5 MAXLOGHISTORY 100
DATAFILE 'diska:dbone.dat' SIZE 2M
MAXDATAFILES 10
MAXINSTANCES 2
ARCHIVELOG
CHARACTER SET US7ASCII NATIONAL
CHARACTER SET JA16SJISFIXED
DATAFILE
'disk1:df1.dbf' AUTOEXTEND ON
'disk2:df2.dbf' AUTOEXTEND ON NEXT 10M MAXSIZE UNLIMITED;
```