

8

L'utilisation de la mémoire par Oracle 10g

Dans ce chapitre :

- l'utilisation générale de la mémoire par Oracle 10g ;
- la mémoire partagée SGA (*System Global Area*) ;
- la mémoire allouée pour chaque programme PGA (*Program Global Area*) ;
- l'espace mémoire de travail des utilisateurs.

Dans tout système informatique, l'utilisation de mémoire est synonyme de performance. Les données auxquelles on accède et qui sont manipulées en mémoire le sont beaucoup plus rapidement que sur disque. C'est pourquoi Oracle 10g, au moyen de techniques efficaces, utilise le plus possible la mémoire physique de vos ordinateurs.

Ce chapitre traite des différents types de mémoires existants et la manière dont Oracle 10g les utilise. Il est important de bien comprendre ces éléments, car ils interviennent dans les opérations d'amélioration des performances.

Mémoire physique et mémoire virtuelle

Un point essentiel consiste à bien définir le type de mémoire dont on parle. Pour ceux qui ne sont pas familiarisés avec ces concepts, rappelons que la mémoire est tout d'abord la mémoire RAM (*Read Access Memory*), c'est-à-dire les circuits installés dans votre ordinateur. C'est le type de mémoire le plus rapide, le plus performant et tous les systèmes d'exploitation l'utilisent.

Néanmoins, la quantité de mémoire RAM n'est pas illimitée. Pour l'augmenter artificiellement, tous les systèmes d'exploitation attribuent à la mémoire RAM un espace disque de débordement. On parle alors de mémoire physique (RAM) et de mémoire virtuelle (espace de *pagination* sous Windows).

La mémoire virtuelle permet de prendre en charge une consommation mémoire plus importante que celle qui tiendrait uniquement dans la mémoire physique. Cela fonctionne particulièrement bien pour les threads qui ne s'activent qu'occasionnellement : leur temps de transfert entre le disque (lieu de stockage) et la mémoire (lieu de traitement) est alors minime.

S'il y a trop de threads actifs en mémoire virtuelle, l'échange continu entre la mémoire physique et virtuelle risque d'engorger votre système. On parle alors de swap ou de pagination du système. Pour pallier la difficulté, il convient soit d'augmenter la mémoire RAM, soit de diminuer la consommation RAM des différents processus.

Le swap transfère un processus entier de la mémoire vers le disque, la pagination ne transfère que quelques pages. Dans un souci de rapidité, les systèmes d'exploitation utilisent massivement la pagination et ils ont recours au swap lorsqu'ils sont engorgés.

Le détail de la consommation mémoire de Windows peut être visualisé de nombreuses façons. L'une des plus simples est d'utiliser le Gestionnaire de tâches.

Mémoire virtuelle sous Windows

Les systèmes d'exploitation Windows 2000, 2003, XP, .NET sont généralement des systèmes 32 bits dont la capacité d'adressage est de 4 Go. Sur la plupart des systèmes, Windows se réserve 2 Go d'espace adresse pour le fonctionnement du système d'exploitation et alloue 2 Go aux processus du système. L'option Microsoft /3GB sur ses serveurs haut de gamme permet de répartir différemment la mémoire : 1 Go pour le système et 3 Go pour les processus utilisateur.

Pour une base de données comme Oracle, cela signifie que chaque processus `oracle.exe` peut adresser un maximum de 2 Go (ou 3 Go avec l'option précitée). La mémoire totale occupée par une instance, sa mémoire partagée et la mémoire privée de tous les utilisateurs connectés, ne pourra excéder cette valeur.

Pour une configuration nécessitant plus de mémoire, Oracle dispose de versions « 64 bits » pour versions « 64 bits » de Windows (2003, XP...). Cette combinaison repousse les limites actuelles des systèmes Windows « 32 bits ».

Toute l'architecture d'Oracle 10g a été conçue en vue d'une utilisation massive de la mémoire physique RAM. Pour des raisons de performance, il est impératif d'éviter que ses différents composants ne se retrouvent dans la zone de mémoire virtuelle.

Les zones mémoire utilisées par Oracle 10g

Comme toute base de données multi-utilisateur, Oracle 10g a besoin de mémoire pour assurer diverses fonctions, à savoir :

- mettre en mémoire, à la disposition de multiples utilisateurs, un maximum d'informations et de données provenant de la base ;
- permettre aux programmes qui gèrent la base Oracle 10g (le processus et ses threads) de fonctionner en mémoire ;
- assurer la transmission des données entre la base Oracle 10g et les threads des utilisateurs ;
- assurer la liaison avec l'application qui utilise les données.

En outre, la mémoire doit accueillir tous les processus nécessaires au fonctionnement de Windows.

Les paramètres statiques et dynamiques

Toutes les bases Oracle utilisent un fichier d'initialisation pour démarrer. Il peut être local (sous forme d'un fichier texte modifiable) ou avoir la forme d'un fichier binaire persistant. Ce point est détaillé au chapitre 23, *Gestion de l'espace disque et des fichiers*.

Parmi les paramètres définis dans le fichier d'initialisation figurent tous ceux qui concernent l'allocation mémoire d'Oracle.

Pour assurer une plus grande disponibilité, les valeurs d'allocation mémoire des instances sont maintenant presque toutes dynamiques, c'est-à-dire qu'elles peuvent être modifiées alors que la base fonctionne :

```
alter system set paramètre=valeur scope= memory ;
```

Les valeurs ainsi fixées peuvent avoir un impact sur la mémoire, être enregistrées dans le fichier d'initialisation persistant ou les deux actions à la fois :

```
alter system set paramètre=valeur scope= memory ;  
alter system set paramètre=valeur scope= spfile ;  
alter system set paramètre=valeur scope= both ;
```

S'il s'agit d'un fichier d'initialisation au format texte, il faut le modifier manuellement pour que la nouvelle valeur soit prise en compte au prochain démarrage. Le fichier d'initialisation persistant est géré directement par Oracle. Il est situé en C:\oracle\product\10.1.0\db_1\database, mais c'est un fichier « caché ». Bien qu'il soit lisible, aucune intervention manuelle n'est permise pour éviter de le corrompre.

Les paramètres d'initialisation dynamiques et ceux qui ne le sont pas sont consultables par la vue V\$PARAMETER.

La zone SGA (System Global Area)

La SGA (*System Global Area*) représente la zone mémoire déterminante d'une instance, tant par sa taille que par son rôle. C'est elle qui assure le partage des données entre les utilisateurs.

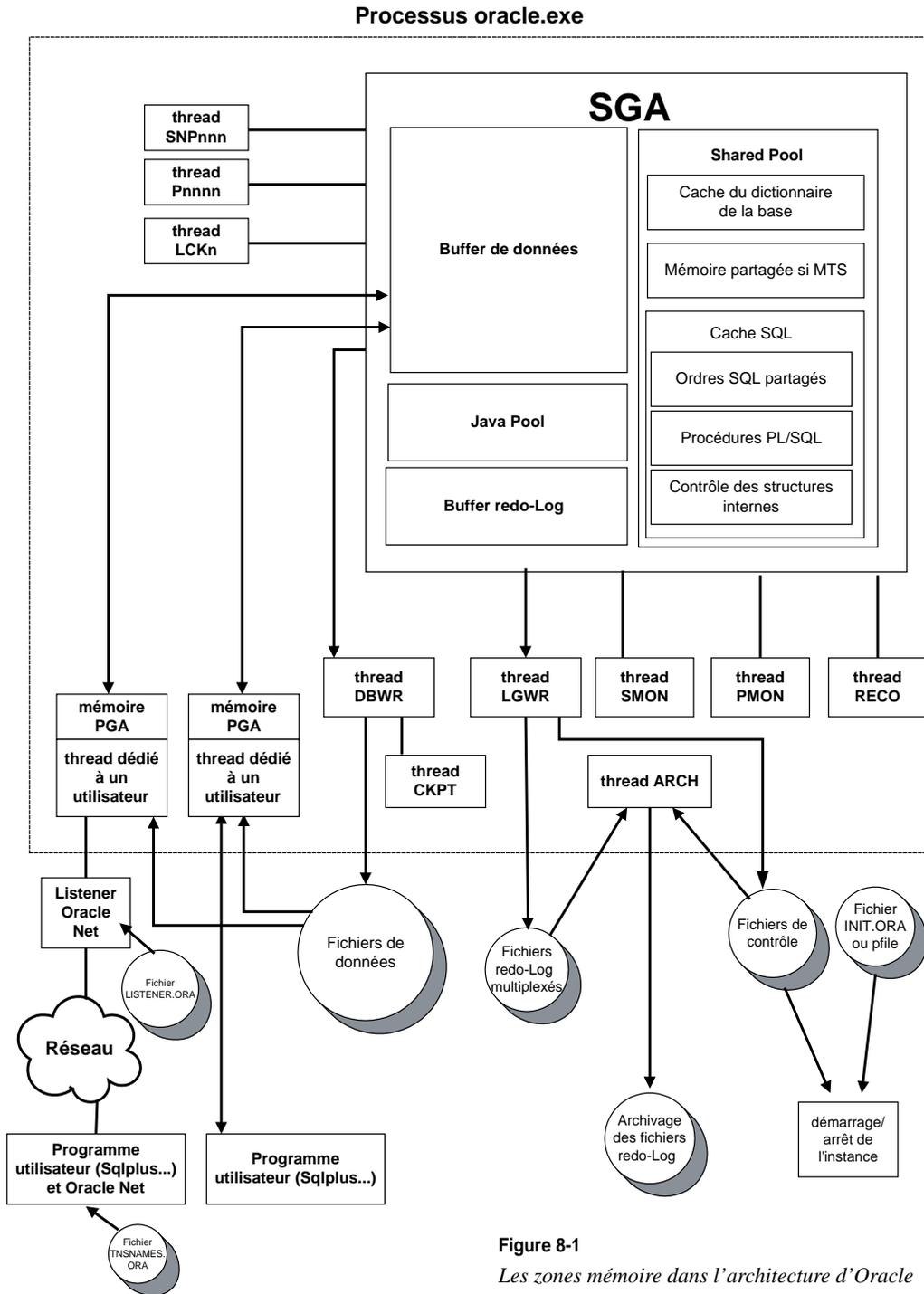


Figure 8-1

Les zones mémoire dans l'architecture d'Oracle

Toute donnée lue ou modifiée transite par la SGA. Même si la totalité du fonctionnement propre à la SGA n'est pas entièrement divulguée par Oracle, il est important d'en comprendre les fonctions majeures, pour faire face à des problèmes de performance ou à un événement inattendu.

Le contenu de la SGA est très structuré. Pour mieux en cerner les différentes zones, nous allons les présenter une par une :

- le cache des blocs de données (*buffer cache*) ;
- les buffers redo-log ;
- la Shared Pool ;
- la Java Pool ;
- une zone de communication inter-processus ;
- des espaces partagés si l'option Multi-Thread est activée.

Le cache des blocs de données

Cette zone comporte toutes les données en provenance de la base ou destinées à y être écrites (données lues par un SELECT ou modifiées par un UPDATE, INSERT, DELETE).

Cette zone mémoire est d'une taille fixe, bien inférieure à la dimension de votre base de données. Il est nécessaire que des mécanismes libèrent de l'espace pour permettre à de nouveaux blocs de « monter » en mémoire. C'est le rôle du processus DBWR et du processus utilisateur.

Le processus DBWR libère de l'espace en écrivant sur disque les blocs de données modifiés. Dans le cas où seuls des blocs de données non modifiés sont présents en mémoire, DBWR ne peut agir. C'est alors qu'intervient le processus utilisateur, créé automatiquement pour chaque connexion à une base Oracle 10g. Ce processus utilisateur obéit à un algorithme qui remplace les blocs non modifiés et inutilisés par des blocs provenant d'une lecture disque. Il lui incombe également de traiter les ordres SQL envoyés par le programme utilisateur et de lui retourner les données.

Comme Oracle 10g n'échange pas des informations mais des blocs d'informations (d'une taille fixe, choisie lors de la création de la base), d'autres données présentes dans ce bloc accompagnent celle à laquelle on a accédé ou qu'on a modifié. Il s'agit là d'un facteur d'amélioration des performances, car ces informations complémentaires seront peut-être utilisées et, dans ce cas, il ne sera pas nécessaire de provoquer un accès disque pour les atteindre.

Nous vous conseillons de consulter le chapitre 28, *Optimisation et performances*, avant de modifier les paramètres d'initialisation d'une instance.

Pour augmenter ou diminuer la taille du cache mémoire, il faut modifier le paramètre d'initialisation DB_CACHE_SIZE. Ce paramètre est dynamique. La valeur affectée est adaptée pour être un multiple du DB_BLOCK_SIZE.

Vous pouvez également utiliser l'ancien paramètre `DB_BLOCK_BUFFER` qui présente l'inconvénient de ne pas être dynamique. Il s'exprime en nombre de blocs mémoire et non en taille mémoire.

Pour visualiser la taille de la SGA, vous pouvez lancer sous le compte administrateur `SYSTEM` la commande SQL :

```
show sga
ou
select * from v$sga;
```

NAME	VALUE
Fixed Size	279600 bytes
Variable Size	167772160 bytes
Database Buffers	67108864 bytes
Redo Buffers	532480 bytes

Total System Global Area	235693104 bytes

Ici, le cache de données (*Database Buffers*) est d'environ 65 Mo pour une SGA de quelques 220 Mo. Cet ordre SQL permet de récupérer la taille des autres zones de la SGA, comme la taille allouée à l'environnement Java ou aux buffers redo-log. Ces valeurs peuvent être modifiées dynamiquement ou dans le fichier d'initialisation.

En interrogeant le dictionnaire de données d'Oracle, d'autres ordres SQL permettent d'entrer plus précisément dans le détail des différentes zones mémoire.

La SGA est un élément clé des performances d'Oracle. Il faut absolument éviter qu'elle « sorte » de la mémoire vive et qu'elle soit paginée. Le paramètre d'initialisation `LOCK_SGA` permet « d'accrocher » la SGA en mémoire vive.

Les paramètres de base et les paramètres avancés

Oracle 10g apporte une simplification de son paramétrage. Il existe maintenant deux types de paramètres :

- les paramètres de base ;
- les paramètres avancés.

Oracle indique que la plupart des bases de données nécessitent uniquement l'utilisation des paramètres de base pour fonctionner correctement. Ces paramètres sont :

```
CLUSTER_DATABASE
COMPATIBLE
CONTROL_FILES
DB_BLOCK_SIZE
DB_CREATE_FILE_DEST
DB_CREATE_ONLINE_LOG_DEST_n
DB_DOMAIN
DB_NAME
```

```
DB_RECOVERY_FILE_DEST
DB_RECOVERY_FILE_DEST_SIZE
DB_UNIQUE_NAME
INSTANCE_NUMBER
JOB_QUEUE_PROCESSES
LOG_ARCHIVE_DEST_n
LOG_ARCHIVE_DEST_STATE_n
NLS_LANG
NLS_TERRITORY
PGA_AGGREGATE_TARGET
PROCESSES
REMOTE_LISTENER
REMOTE_LOGIN_PASSWORDFILE
ROLLBACK_SEGMENTS
SESSIONS
SGA_TARGET
SHARED_SERVERS
START_TRANSFORMATION_ENABLED
UNDO_MANAGEMENT
UNDO_TABLESPACE
```

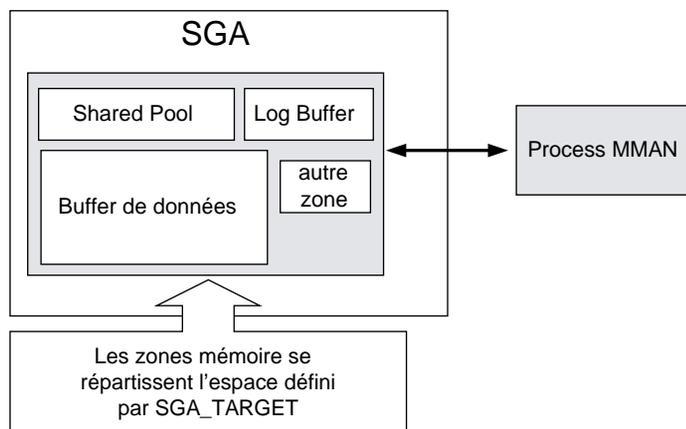
Consultez l'index de cet ouvrage pour identifier la page où sont abordés ces différents paramètres. Pour les autres, reportez-vous à la documentation Oracle.

Le tuning automatique de la SGA

Oracle 10g apporte une nouveauté : l'Automatic Shared Memory Management (ASMM) et son processus Memory Manager (MMAN). Il laisse Oracle répartir automatiquement un espace mémoire disponible, le **sga_target** entre le buffer de données, la Shared Pool, la Large Pool et la Java Pool. C'est une fonction très utile pour des bases dont l'activité variable peut charger temporairement l'une de ces zones mémoire.

Ce paramètre est situé dans le fichier init.ora. Sa valeur par défaut est **sga_target = 0** qui inhibe le fonctionnement de l'ASMM.

Figure 8-2
Le principe d'ASMM



Pour utiliser ASMM, il suffit :

- de donner une taille mémoire à `sga_target` (autre que 0). Vous pouvez vous baser sur le résultat de l'ordre SQL présenté dans le paragraphe précédent ;
- de positionner le paramètre `statistics_level` à `TYPICAL` ou `ALL` ;
- de positionner les valeurs définissant les buffers de données, la Shared Pool, la Java Pool, etc., à 0. Si une valeur autre que 0 est conservée, ASMM l'interprétera comme une taille minimale de mémoire à conserver pour cette zone.

Le paramètre `sga_target` est dynamique, c'est-à-dire qu'il peut être changé base ouverte, en fonctionnement. Veuillez toutefois à ne pas lui allouer un espace mémoire dont vous ne disposez pas !

Les buffers redo-log

Dès qu'une modification intervient sur vos données (INSERT, UPDATE, DELETE), elle se répercute sur les blocs de données en mémoire, qui sont simultanément copiés dans les buffers redo-log.

Dès qu'un COMMIT ou un ROLLBACK est utilisé, le processus LGWR écrit les données (non encore transférées) dans les fichiers redo-log. L'utilisateur n'a pas à attendre l'écriture des éléments nouveaux dans les fichiers de données (ce qui peut être long). Ce travail est effectué en temps masqué par le processus DBWR. Aucune information ne sera perdue, quoi qu'il arrive, car toutes sont écrites dans les fichiers redo-log.

Les buffers redo-log sont conçus pour assurer des écritures très rapides dans les fichiers redo-log. Comme pour tous les autres échanges, ce sont des blocs de données qui sont écrits dans ces fichiers, ce processus étant le plus performant.

Le paramètre dynamique LOG_BUFFER dimensionne cette zone mémoire. Il s'exprime en octets et il peut être modifié dynamiquement.

La Shared Pool

Nous avons vu comment les buffers de données peuvent améliorer l'accès aux données. Néanmoins, une base de données ne comporte pas que les informations de l'utilisateur, elle inclut également toutes celles nécessaires à son fonctionnement. Là aussi, des mécanismes ont été construits pour fournir d'excellentes performances. Pour cela, les zones mémoire utilisées sont :

- le partage des ordres SQL et PL/SQL : Shared SQL Area ;
- le dictionnaire de données ;
- les curseurs ;
- le Multi-Thread.

Le paramètre dynamique `SHARED_POOL_SIZE` dimensionne cette zone mémoire. Il s'exprime en octets et il peut être modifié dynamiquement.

Le langage SQL utilisé pour dialoguer avec Oracle 10g doit être traduit et interprété par la base pour vous apporter une réponse. À cette fin, une analyse syntaxique (la syntaxe est-elle correcte ?) et sémantique (tous les éléments sont-ils compris ?) est mise en œuvre. Oracle 10g vérifie aussi que vous disposez des droits d'accès aux données demandées. Enfin, Oracle 10g consulte ses statistiques (en tenant compte du volume de vos tables et des index) pour optimiser le chemin d'accès à vos données. Cette opération correspond au *parsing* de votre ordre SQL. Tout ce travail est rapide, mais quand il s'agit d'applications qui répètent très souvent le même ordre SQL ou des ordres proches, certains éléments sont conservés et partagés en vue d'une réutilisation.

Ce mécanisme est particulièrement efficace dans le traitement des programmes PL/SQL. Le code d'exécution de vos procédures, fonctions, triggers et packages est conservé, prêt à être réemployé.

Lors de l'analyse des ordres SQL, Oracle 10g vérifie le nom des tables, des colonnes, les droits d'accès, etc. Tous ces éléments sont gérés dans le dictionnaire de la base de données, propriété de l'utilisateur SYS. Une zone mémoire est allouée au stockage d'informations provenant du dictionnaire de données.

Les curseurs sont utilisés dans les programmes PL/SQL et dans des opérations internes réalisées par Oracle. Tout comme pour les ordres SQL et les programmes PL/SQL, un espace mémoire est réservé en Shared Pool pour les curseurs.

La Java Pool

Oracle 10g propose un environnement Java intégré à la base. Il nécessite de la mémoire pour fonctionner. L'installation du moteur Java dans la base Oracle est facultative. Ce point est abordé au chapitre 13, *Création d'une base Oracle 10g*.

La mémoire allouée en SGA pour le fonctionnement du moteur Java est précisée par la variable `JAVA_POOL_SIZE`. Cette valeur est fixée à 50 Mo minimum lors de l'installation du catalogue Java, mais elle peut être augmentée ou diminuée par la suite.

La zone PGA (Program Global Area)

Le rôle de la SGA est de permettre un partage de données entre tous les utilisateurs. Il existe néanmoins d'autres informations qui n'ont pas besoin d'être partagées. C'est le cas des informations contenues dans la zone PGA (*Program Global Area*, parfois appelée *Process Global Area*). Cette zone mémoire est allouée pour le fonctionnement de chaque thread utilisateur.

Une connexion à une base Oracle 10g engendre donc la création d'un thread utilisateur et de sa mémoire associée, la PGA.

La PGA est toujours située sur le serveur qui héberge votre base de données. Elle stocke des informations concernant les variables utilisées, la session utilisateur et l'état des transactions en cours. Elle contient également la zone mémoire dans laquelle s'effectue le tri de vos données.

Lorsque de nombreux utilisateurs accèdent à cette base, la multiplication des processus utilisateur et de chacune de leur PGA conduit à une surcharge du système. C'est à ce moment que l'utilisation de l'option MTS (*Multi-Thread Server*) peut économiser des ressources en transférant une partie des opérations effectuées dans la PGA vers la SGA. C'est le cas de la zone de tri mémoire de vos ordres SQL. L'inconvénient majeur de cette technique est la forte augmentation de la SGA. Il convient donc de trouver un équilibre entre une « petite » SGA et des PGA « consommatrices », et une SGA « imposante » et des PGA « plus petites ».

Le choix d'une configuration MTS se justifie à partir de cent ou cent cinquante utilisateurs simultanés.

La taille de la PGA est déterminée par des paramètres d'initialisation de votre instance : la taille mémoire prévue pour le tri des données, le nombre de « database links » autorisés, le nombre de fichiers composant votre base de données et le nombre de fichiers redolog. En ce qui concerne l'espace de tri des données, il faut bien réaliser l'importance du paramètre d'initialisation SORT_AREA_SIZE qui influe sur la taille de toutes les PGA créées.

Depuis Oracle 10g, la gestion de la taille de la PGA peut être automatique : PGA_AGGREGATE_TARGET donne la mémoire maximale que vous allouez pour l'ensemble des PGA utilisateurs. Cette zone ne réside pas en SGA. La valeur WORKAREA_SIZE_POLICY peut prendre les valeurs *manual* ou *auto*.

L'administrateur d'une base Oracle 10g n'a pas à se soucier de l'existence et du fonctionnement de la PGA (c'est pourquoi elle est si peu connue !). Il convient d'y songer lorsque vous rencontrez des limites de mémoire sur votre système : vous pouvez alors augmenter sa capacité, diminuer certains paramètres ou envisager une configuration MTS.

Résumé

Ce chapitre a détaillé les différentes zones mémoire utilisées par Oracle 10g. Elles sont partagées ou privées et vous permettent d'accéder le plus rapidement possible à vos données. La compréhension de leur fonctionnement pour les dimensionner est très importante : c'est l'un des facteurs clés de l'amélioration des performances de vos bases Oracle 10g.