

Oracle Database 11g : Administration Workshop II

Manuel du stagiaire - Volume II

D50079FR20

Edition 2.0

Septembre 2010

D67999

ORACLE

Auteur

Maria Billings

Révisions et contributions techniques

Christian Bauwens

Yanti Chang

Timothy Chien

Joe Fong

Andy Fortunak

Gerlinde Frenzen

Mark Fuller

Peter Fusek

Joel Goodman

Vimala Jacob

Dominique Jeunot

Pete Jones

Fukue Kawabe

Donna Keesling

Sean Kim

Achiel Langers

Gwen Lazenby

Jerry Lee

Deirdre Matishak

Bill Millar

Lakshmi Naraparreddi

Ira Singer

Ranbir Singh

James Spiller

Matt Taylor

Branislav Valny

Jean-François Verrier

Rédacteurs

Nita Pavitran

Raj Kumar

Concepteur graphique

Satish Bettgowda

Editeur

Jayanthi Keshavamurthy

Copyright © 2009, Oracle. Tous droits réservés.

Avertissement

Cette documentation contient des informations qui sont la propriété d'Oracle Corporation et sont protégées par les lois relatives aux droits d'auteur et à la propriété intellectuelle. Vous ne pouvez copier et imprimer ce document qu'à des fins d'utilisation personnelle lors de la participation à une formation dispensée par Oracle. Le document ne peut être modifié ou altéré en aucune manière. A l'exception des cas où l'utilisation faite du document s'inscrit dans le respect des lois relatives aux droits d'auteur, vous ne pouvez pas utiliser, partager, télécharger, copier, imprimer, afficher, exécuter, reproduire, publier, breveter, diffuser, transmettre ou distribuer ce document, en partie ou en totalité, sans l'autorisation expresse d'Oracle.

Les informations fournies dans ce document sont susceptibles de modification sans préavis. Par ailleurs, Oracle Corporation ne garantit pas qu'elles soient exemptes d'erreurs et vous invite, le cas échéant, à lui en faire part par écrit à l'adresse suivante : Oracle University, 500 Oracle Parkway, Redwood Shores, California 94065 USA.

Restrictions applicables au gouvernement américain : Restricted Rights Notice

If this documentation is delivered to the United States Government or anyone using the documentation on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

The U.S. Government's rights to use, modify, reproduce, release, perform, display, or disclose these training materials are restricted by the terms of the applicable Oracle license agreement and/or the applicable U.S. Government contract.

Marques

Oracle est une marque déposée d'Oracle Corporation et/ou de ses filiales. Tout autre nom de produit ou de société peut être une marque de son propriétaire.

Table des matières

I Introduction

- Objectifs du cours 1-2
- Planning suggéré 1-3
- Oracle Database 11g : "g" signifie "grid" 1-4
- Grid Infrastructure for Single-Instance 1-6
- Exemples du cours : Exemple de schéma HR 1-8

1 Concepts de base et outils d'Oracle Database

- Objectifs 1-2
- Noms des composants élémentaires d'un serveur Oracle Database 1-3
- Architecture d'un serveur de base de données Oracle 1-4
- Configurations instance-base de données 1-6
- Noms des structures mémoire d'une base de données Oracle 1-7
- Structures mémoire d'une base de données Oracle 1-8
- Architecture de processus 1-10
- Structures de processus 1-11
- Exercice sur les noms de processus 1-13
- Séquence de démarrage des processus 1-14
- Architecture de stockage de la base de données 1-15
- Structures logiques et structures physiques d'une base de données 1-17
- Automatic Storage Management 1-19
- Composants de stockage ASM 1-20
- Instance ASM 1-21
- Outils de configuration destinés aux DBA 1-23
- Environnement de gestion et outils associés pour les DBA 1-24
- Faciliter la gestion de la base de données avec Oracle Restart 1-25
- Quiz 1-27
- Synthèse 1-28

2 Configurer la base de données afin d'optimiser la possibilité de récupération

- Objectifs 2-2
- Finalité de la fonction de sauvegarde et de récupération 2-3
- Opérations classiques de sauvegarde et de récupération 2-4
- Solutions de sauvegarde et de récupération Oracle 2-5
- Solutions de sauvegarde Oracle 2-6
- Terminologie - Exercice 2-7
- Rappels : Sauvegarde recommandée par Oracle 2-9
- Utiliser Recovery Manager 2-10
- Types de commande RMAN 2-12

Commandes de type travail : Exemple 2-13
Configurer la base de données pour des opérations de sauvegarde
et de récupération 2-14
Mode ARCHIVELOG 2-15
Configurer le mode ARCHIVELOG 2-16
Configurer des destinations de fichiers de journalisation archivés 2-17
Garantir la réussite des fichiers de journalisation archivés 2-18
Indiquer une stratégie de conservation 2-20
Indiquer une stratégie de conservation avec fenêtre
de récupération : Exemple 2-22
Utiliser une zone de récupération rapide 2-23
Définir une zone de récupération rapide 2-25
Gestion de l'espace dans la zone de récupération rapide 2-26
Opérations réalisées automatiquement 2-30
Surveiller la zone de récupération rapide 2-31
Avantages de l'utilisation d'une zone de récupération rapide 2-32
Quiz 2-33
Synthèse 2-35
Présentation de l'exercice 2 : Configurer la base de données afin d'optimiser
la possibilité de récupération 2-36

3 Utiliser le catalogue de restauration RMAN

Objectifs 3-2
Stockage des données du référentiel RMAN : Comparaison des options 3-3
Stocker des informations dans le catalogue de restauration 3-4
Utilité du catalogue de restauration 3-5
Créer le catalogue de restauration : Trois étapes 3-6
Configurer la base de données du catalogue de restauration 3-7
Créer le propriétaire du catalogue de restauration 3-8
Créer le catalogue de restauration 3-9
Gérer les enregistrements de la base de données cible dans le catalogue
de restauration 3-10
Enregistrer une base de données dans le catalogue de restauration 3-11
Utiliser Enterprise Manager pour enregistrer une base de données 3-12
Supprimer l'enregistrement d'une base de données cible dans le catalogue
de restauration 3-13
Enregistrer des fichiers de sauvegarde supplémentaires dans le catalogue 3-14
Resynchronisation du catalogue de restauration : Concepts 3-16
Resynchroniser manuellement le catalogue de restauration 3-17
Utiliser des scripts RMAN stockés 3-18
Exécuter des scripts RMAN stockés 3-19
Gérer les scripts RMAN stockés 3-20
Sauvegarder le catalogue de restauration 3-21
Recréer un catalogue de restauration irrécupérable 3-22

Exporter et importer le catalogue de restauration 3-23
Mettre à niveau et supprimer le catalogue de restauration 3-24
Commande `IMPORT CATALOG` 3-26
Créer et utiliser des catalogues privés virtuels 3-28
Utiliser les catalogues privés virtuels RMAN 3-29
Récapitulatif sur les catalogues de restauration 3-31
Quiz 3-33
Synthèse 3-35
Présentation de l'exercice 3 : Utiliser le catalogue de restauration RMAN 3-36

4 Configurer les paramètres de sauvegarde

Objectifs 4-2
Configurer des paramètres persistants pour RMAN 4-3
Visualiser les paramètres persistants 4-4
Sauvegarde automatique du fichier de contrôle 4-5
Gérer les paramètres persistants 4-7
Utiliser un gestionnaire de support 4-8
Indiquer une destination de sauvegarde 4-10
Configurer et allouer des canaux 4-12
Créer des jeux de sauvegarde multiplexés 4-13
Créer des jeux de sauvegarde multiplexés à l'aide
de la commande `CONFIGURE BACKUP COPIES` 4-14
Optimisation de la sauvegarde 4-15
Economiser de l'espace de sauvegarde par compression
des blocs inutilisés 4-16
Compresser des sauvegardes 4-17
Utiliser la compression RMAN des sauvegardes 4-18
Crypter des sauvegardes 4-19
Quiz 4-20
Synthèse 4-22
Présentation de l'exercice 4 : Configurer les spécifications de sauvegarde 4-23

5 Créer des sauvegardes avec RMAN

Objectifs 5-2
Créer des jeux de sauvegarde 5-3
Créer des copies d'image 5-4
Créer une sauvegarde totale de la base de données 5-6
Types de sauvegarde RMAN 5-8
Sauvegarde incrémentielle rapide 5-10
Activer une sauvegarde incrémentielle rapide 5-11
Surveiller le suivi des modifications de blocs 5-12
Réaliser des "proxy copies" (copies déléguées à un système tiers) 5-13
Créer des jeux de sauvegarde multiplexés à l'aide
de la commande `BACKUP COPIES` 5-14

Créer des sauvegardes de jeux de sauvegarde	5-15
Sauvegarder des tablespaces en lecture seule	5-16
Configurer la sauvegarde et la restauration pour les fichiers très volumineux	5-17
Créer des sauvegardes RMAN multisections	5-18
Sauvegardes d'archivage : Concepts	5-19
Créer des sauvegardes d'archivage avec EM	5-21
Créer des sauvegardes d'archivage avec RMAN	5-22
Gérer les sauvegardes d'archivage de la base de données	5-23
Sauvegarder des fichiers de récupération	5-24
Gérer les sauvegardes : Créer des états	5-25
Gérer les sauvegardes : Vues dynamiques des performances	5-27
Utiliser Enterprise Manager pour afficher les états de sauvegarde	5-28
Gérer les sauvegardes : Vérification croisée et suppression	5-29
Quiz	5-30
Synthèse	5-32
Présentation de l'exercice 5 : Créer des sauvegardes	5-33

6 Opérations de restauration et de récupération

Objectifs	6-2
Restauration et récupération	6-3
Causes possibles de la perte de fichiers	6-4
Perte d'un fichier non critique	6-5
Récupération automatique d'un fichier Tempfile	6-6
Statut d'un groupe de fichiers de journalisation : Rappel	6-7
Récupération suite à la perte d'un membre du groupe de fichiers de journalisation	6-8
Vider un fichier journal	6-9
Récupération suite à la perte d'un tablespace d'index	6-10
Recréer des index	6-11
Méthodes d'authentification pour les administrateurs de base de données	6-13
Recréer un fichier d'authentification par mot de passe	6-14
Récupération complète et récupération incomplète	6-16
Processus de récupération complète	6-17
Récupération jusqu'à un point dans le temps	6-18
Récupérer un tablespace en lecture seule	6-20
Récupérer des objets de base de données NOLOGGING	6-21
Récupération suite à la perte de toutes les copies du fichier de contrôle : Présentation	6-22
Récupérer le fichier de contrôle à l'emplacement par défaut	6-23
Quiz	6-24
Synthèse	6-26

7 Utiliser RMAN pour procéder à une récupération

Objectifs 7-2

Utiliser les commandes RMAN RESTORE et RECOVER 7-3

Procéder à une récupération complète : Perte d'un fichier de données non critique en mode ARCHIVELOG 7-4

Procéder à une récupération complète : Perte d'un fichier de données critique pour un système en mode ARCHIVELOG 7-5

Récupérer des copies d'image 7-6

Récupérer des copies d'image : Exemple 7-7

Basculement rapide vers des copies d'image 7-8

Utiliser la commande SET NEWNAME pour changer de fichier 7-9

Variables de substitution pour SET NEWNAME 7-10

Procéder à la restauration et à la récupération d'une base de données en mode NOARCHIVELOG 7-11

Utiliser des points de restauration 7-12

Effectuer une récupération jusqu'à un point dans le temps 7-13

Procéder à une récupération avec un fichier de contrôle de sauvegarde 7-15

Récupération suite à la perte du fichier de paramètres serveur 7-16

Restaurer le fichier de paramètres serveur à partir de la sauvegarde automatique du fichier de contrôle 7-17

Restaurer le fichier de contrôle à partir de la sauvegarde automatique 7-18

Utiliser des sauvegardes incrémentielles pour récupérer une base de données en mode NOARCHIVELOG 7-20

Restaurer et récupérer la base de données sur un nouvel hôte 7-21

Préparation de la restauration de la base de données sur un nouvel hôte 7-22

Restaurer la base de données sur un nouvel hôte 7-23

Procéder à une récupération après sinistre 7-27

Quiz 7-29

Synthèse 7-31

Présentation de l'exercice 7 : Utiliser RMAN pour procéder à une récupération 7-32

8 Surveiller et régler RMAN

Objectifs 8-2

Exécution en parallèle de jeux de sauvegarde 8-3

Surveiller les sessions RMAN 8-5

Surveiller la progression des travaux RMAN 8-7

Interpréter les messages RMAN 8-9

Utiliser l'option DEBUG 8-10

Interpréter les piles d'erreur RMAN 8-11

Régler RMAN 8-12

Multiplexage RMAN 8-14

Allouer des mémoires tampons sur disque : Exemple 8-15

- Allouer des mémoires tampons sur bande 8-16
- Comparer les E/S synchrones et asynchrones 8-18
- Surveiller les performances des travaux RMAN 8-20
- Goulets d'étranglement avec E/S asynchrones 8-21
- Goulets d'étranglement avec E/S synchrones 8-22
- Réglage des canaux 8-23
- Régler la commande `BACKUP` 8-25
- Régler les performances de sauvegarde RMAN 8-27
- Définir le paramètre `LARGE_POOL_SIZE` 8-28
- Régler les goulets d'étranglement affectant les performances de la transmission en continu sur bande RMAN 8-30
- Quiz 8-32
- Synthèse 8-34
- Présentation de l'exercice 8 : Surveiller et régler RMAN 8-35

9 Diagnostiquer la base de données

- Objectifs 9-2
- Data Recovery Advisor 9-3
- Défaillances de données 9-6
- Défaillances de données : Exemples 9-7
- Data Recovery Advisor : Interface de ligne de commande RMAN 9-8
- Lister les défaillances de données 9-9
- Conseils sur la réparation 9-11
- Exécuter des réparations 9-12
- Classer (et fermer) les défaillances 9-13
- Vues de Data Recovery Advisor 9-14
- Méthode recommandée : Vérifications proactives 9-15
- Qu'est-ce qu'une corruption de bloc ? 9-16
- Symptômes d'une corruption de bloc : `ORA-01578` 9-17
- Comment traiter une corruption 9-18
- Définir les paramètres pour la détection des corruptions 9-19
- Restauration physique de bloc (BMR) 9-21
- Prérequis à la restauration physique de bloc 9-22
- Commande `RECOVER...BLOCK` 9-23
- Workflow de diagnostic automatique 9-24
- Référentiel ADR 9-25
- ADRCI : Outil de ligne de commande du référentiel ADR 9-26
- Vue `V$DIAG_INFO` 9-27
- Emplacement des traces de diagnostic 9-28
- Health Monitor : Présentation 9-29
- Exécuter manuellement des vérifications de l'état général :
 - Exemple PL/SQL 9-30
- Consulter des états HM à l'aide de l'utilitaire ADRCI 9-31
- Quiz 9-32

Synthèse 9-36

Présentation de l'exercice 9 : Diagnostiquer la base de données 9-37

10 Utiliser la technologie Flashback I

Objectifs 10-2

Technologie Flashback 10-3

Transactions et informations d'annulation 10-4

Garantir la période de conservation des informations d'annulation 10-5

Préparer la base de données pour un flashback 10-6

Utiliser la technologie Flashback pour interroger des données 10-8

Flashback Query 10-9

Flashback Query : Exemple 10-10

Flashback Version Query 10-11

Flashback Version Query : Eléments à prendre en compte 10-12

Quiz 10-13

Flashback Table : Présentation 10-15

Flashback Table 10-16

Activer le déplacement de lignes (row movement) dans une table 10-17

Procéder au flashback d'une table 10-18

Flashback Table : Eléments à prendre en compte 10-19

Quiz 10-20

Flashback Transaction Query 10-21

Utiliser Enterprise Manager pour exécuter une opération

Flashback Transaction Query 10-22

Flashback Transaction Query : Eléments à prendre en compte 10-23

Flashback Transaction 10-24

Prérequis 10-25

Procéder au flashback d'une transaction 10-26

Workflow possible 10-27

Assistant Flashback Transaction Wizard 10-28

Choisir d'autres options d'annulation 10-29

Etapes finales sans EM 10-31

Quiz 10-32

Synthèse 10-33

Présentation de l'exercice 10 : Effectuer une annulation à l'aide de

Flashback Transaction 10-34

11 Utiliser la technologie Flashback II

Objectifs 11-2

Présentation d'Oracle Total Recall 11-3

Processus de configuration 11-5

Fonctionnement d'Oracle Total Recall 11-6

Scénario utilisant Oracle Total Recall 11-7

Evolution transparente de schéma 11-10

Evolution complète de schéma 11-11
Restrictions 11-12
Règles 11-13
Consulter des Flashback Data Archives 11-14
Quiz 11-15
Flashback Drop et la corbeille 11-17
Corbeille 11-18
Restaurer des tables à partir de la corbeille 11-20
Corbeille : Récupération automatique d'espace 11-21
Corbeille : Récupération manuelle d'espace 11-22
Contourner la corbeille 11-23
Interroger la corbeille 11-24
Quiz 11-25
Synthèse 11-26
Présentation de l'exercice 11 : Utiliser la technologie Flashback 11-27

12 Utiliser Flashback Database

Objectifs 12-2
Flashback Database 12-3
Architecture Flashback Database 12-4
Configurer Flashback Database 12-5
Opérations à effectuer 12-6
Flashback Database : Exemples 12-7
Considérations relatives à Flashback Database 12-8
Surveiller Flashback Database 12-9
Surveiller Flashback Database avec EM 12-11
Points de restauration garantis 12-12
Flashback Database et points de restauration garantis 12-13
Quiz 12-15
Synthèse 12-17
Présentation de l'exercice 12 : Utiliser Flashback Database 12-18

13 Gérer la mémoire

Objectifs 13-2
Gestion de la mémoire : Présentation 13-3
Rappels sur les structures mémoire d'une base Oracle 13-4
Cache de tampons 13-6
Utiliser des pools de tampons multiples 13-8
Zone de mémoire partagée 13-10
Zone de mémoire LARGE POOL 13-11
Zones de mémoire Java et Streams 13-12
Tampon de journalisation 13-13
Gestion automatique de la mémoire : Présentation 13-14
Paramètres de dimensionnement de la mémoire Oracle Database 13-15

Surveiller la gestion automatique de la mémoire	13-16
Utilisation efficace de la mémoire : Recommandations	13-18
Recommandations relatives au réglage de la mémoire pour le cache "library"	13-20
Gestion automatique de la mémoire partagée : Présentation	13-22
Fonctionnement de la gestion automatique de la mémoire partagée	13-23
Activer la gestion automatique de la mémoire partagée	13-24
Désactiver la fonction ASMM	13-25
Mémoire PGA	13-26
Utiliser la vue V\$PARAMETER	13-28
Quiz	13-29
Synthèse	13-30
Présentation de l'exercice 13 : Utiliser la fonction AMM pour corriger un problème d'allocation de mémoire	13-31

14 Gérer les performances de la base de données

Objectifs	14-2
Activités de réglage	14-3
Planification des performances	14-4
Réglage de l'instance	14-6
Méthodologie de réglage des performances	14-7
Surveillance des performances	14-8
Données de réglage des performances	14-9
Collecte des statistiques destinées à l'optimiseur	14-10
Préférences relatives aux statistiques : Présentation	14-12
Utiliser les préférences relatives aux statistiques	14-13
Configurer les préférences globales avec Enterprise Manager	14-14
Evénements Wait Oracle	14-15
Statistiques au niveau instance	14-16
Surveiller les performances des sessions	14-19
Afficher les statistiques liées aux sessions	14-20
Afficher les statistiques liées aux services	14-21
Vues de dépannage et de réglage	14-22
Vues du dictionnaire	14-23
Référentiel AWR	14-24
Utiliser les vues du référentiel AWR	14-26
Présentation de Real Application Testing : Database Replay	14-27
Vue d'ensemble	14-28
Quiz	14-29
Synthèse	14-30
Présentation de l'exercice 14 : Surveiller les performances d'une instance	14-31

15 Gérer les performances via le réglage des instructions SQL

- Objectifs 15-2
- Réglage des instructions SQL 15-3
- Fonctions de conseil SQL 15-4
- Résultats du réglage automatique des instructions SQL 15-6
- Implémenter les recommandations de réglage automatique 15-7
- SQL Tuning Advisor : Présentation 15-8
- Utiliser SQL Tuning Advisor 15-9
- Options de la fonction de conseil SQL Tuning Advisor 15-10
- Recommandations SQL Tuning Advisor 15-11
- Utiliser SQL Tuning Advisor : Exemple 15-12
- Instructions SQL en double 15-13
- SQL Access Advisor : Présentation 15-14
- Session SQL Access Advisor typique 15-15
- Source de la charge globale 15-16
- Options relatives aux recommandations 15-17
- Examiner les recommandations 15-19
- SQL Performance Analyzer : Présentation 15-20
- SQL Performance Analyzer : Cas d'utilisation 15-21
- Utiliser SQL Performance Analyzer 15-22
- Quiz 15-23
- Synthèse 15-27
- Présentation de l'exercice 15 : Régler les instructions SQL
pour améliorer les performances 15-28

16 Gérer les ressources

- Objectifs 16-2
- Database Resource Manager : Présentation 16-3
- Database Resource Manager : Concepts 16-4
- Pourquoi utiliser Resource Manager 16-5
- Plan Resource Manager de maintenance par défaut 16-8
- Exemple : DEFAULT_PLAN 16-9
- Workflow possible 16-10
- Définir les directives du plan d'allocation de ressources 16-12
- Méthodes d'allocation des ressources pour les plans d'allocation
de ressources 16-13
- Comparaison entre EMPHASIS et RATIO 16-15
- Mécanisme du pool de sessions actives 16-17
- Configurer le pool de sessions actives 16-18
- Définir des seuils 16-20
- Définir des délais d'inactivité 16-21
- Limiter l'utilisation de la CPU au niveau base de données 16-22
- Limiter l'utilisation de CPU au niveau serveur : Mise en cage d'instance 16-24
- Exemples de mise en cage d'instance 16-25

Surveiller la mise en cage d'instance 16-26
Mapping des groupes de consommateurs de ressources 16-27
Activer un plan d'allocation de ressources 16-29
Informations relatives à Database Resource Manager 16-30
Surveiller Resource Manager 16-31
Quiz 16-34
Synthèse 16-35
Présentation de l'exercice 16 : Utiliser Resource Manager 16-36

17 Automatiser des tâches avec le planificateur

Objectifs 17-2
Simplifier les tâches de gestion 17-3
Composants essentiels 17-4
Workflow de base 17-5
Quiz 17-7
Travaux légers persistants 17-8
Utiliser une planification basée sur une date/heure ou sur les événements 17-9
Créer un travail basé sur une date/heure 17-10
Créer une planification basée sur les événements 17-12
Créer des planifications basées sur les événements
avec Enterprise Manager 17-13
Créer un travail basé sur les événements 17-14
Planification basée sur les événements 17-15
Créer une planification complexe 17-17
Quiz 17-18
Utiliser les notifications par e-mail 17-19
Ajouter et supprimer des notifications par e-mail 17-20
Créer des chaînes de travaux 17-21
Exemple de chaîne 17-23
Fonctionnalités avancées du planificateur 17-24
Classes de travaux 17-25
Fenêtres 17-27
Affecter des priorités aux travaux dans une fenêtre 17-28
Créer un ensemble de travaux 17-29
Quiz 17-31
Créer un contrôleur de fichier et un travail fondé sur les événements 17-32
Activer les événements relatifs à l'arrivée du fichier
à partir de systèmes distants 17-34
Planifier des travaux sur une base de données distants 17-35
Créer des travaux sur une base de données distante 17-36
Planifier des travaux à destinations multiples 17-37
Afficher les métadonnées du planificateur 17-38
Quiz 17-40
Synthèse 17-41

Présentation de l'exercice 17 : Automatiser des tâches
avec le planificateur 17-42

18 Gérer l'espace

Objectifs 18-2

Gestion de l'espace : Présentation 18-3

Gestion de l'espace de bloc 18-4

Chaînage et migration de lignes 18-5

Quiz 18-7

Gestion de l'espace libre dans des segments 18-8

Types de segment 18-9

Allocation d'extents 18-10

Allouer de l'espace 18-11

Créer des tables sans segments 18-12

Contrôler la création différée de segments 18-13

Restrictions et exceptions 18-14

Fonctionnalités automatiques supplémentaires 18-15

Quiz 18-16

Compression de table : Présentation 18-17

Compression pour les insertions par chemin direct 18-18

Compression OLTP pour les opérations LMD 18-20

Définir la compression de table 18-21

Utiliser la fonction de conseil sur la compression 18-22

Utiliser le package DBMS_COMPRESSION 18-23

Compression des données d'une table 18-24

Surveillance proactive des tablespaces 18-25

Seuils et résolution de problèmes d'espace 18-26

Surveillance de l'utilisation de l'espace dans les tablespaces 18-27

Récupération d'espace dans les segments 18-28

Résultats d'une opération de récupération d'espace 18-29

Récupérer de l'espace dans des segments ASSM 18-30

Segment Advisor : Présentation 18-31

Segment Advisor 18-32

Implémenter les recommandations 18-33

Automatic Segment Advisor 18-34

Récupération manuelle d'espace dans les segments à l'aide d'EM 18-35

Récupération d'espace dans les segments à l'aide d'instructions SQL 18-36

Gérer la reprise après un problème d'allocation d'espace 18-38

Utiliser le mode de reprise après un problème d'allocation d'espace 18-40

Reprise d'instructions mises en suspens 18-42

Opérations pouvant faire l'objet d'une reprise 18-44

Quiz 18-45

Synthèse 18-46

Présentation de l'exercice 18 : Gérer le stockage 18-47

19 Gérer l'espace de la base de données

- Objectifs 19-2
- Structures de stockage de la base de données 19-3
- Prise en charge des disques avec secteurs de 4 ko 19-4
- Utiliser des disques avec secteurs de 4 ko 19-5
- Définir la taille de secteur d'un disque 19-6
- Quiz 19-7
- Transport de tablespaces 19-10
- Concept : Niveau de compatibilité minimum 19-11
- Niveau de compatibilité minimum 19-12
- Procédure relative aux tablespaces transportables 19-13
- Déterminer le "endian format" d'une plate-forme 19-14
- Utiliser la commande RMAN CONVERT 19-16
- Tablespaces transportables avec Enterprise Manager 19-17
- Transport de base de données 19-20
- Procédure de transport de base de données : Conversion du système source 19-21
- Procédure de transport de base de données : Conversion du système cible 19-22
- Transport d'une base de données : Considérations 19-23
- Quiz 19-24
- Synthèse 19-25
- Présentation de l'exercice 19 : Gérer l'espace de la base de données 19-26

20 Dupliquer une base de données

- Objectifs 20-2
- Utiliser une base de données dupliquée 20-3
- Choisir la technique de duplication 20-4
- Dupliquer une base de données active 20-5
- Dupliquer une base de données avec une connexion à la cible 20-6
- Dupliquer une base de données à l'aide du catalogue de restauration sans connexion à la cible 20-7
- Dupliquer une base de données sans catalogue de restauration ni connexion à la cible 20-8
- Créer une base de données dupliquée à partir de sauvegardes 20-9
- Créer un fichier de paramètres d'initialisation pour l'instance auxiliaire 20-10
- Indiquer de nouveaux noms pour la destination 20-11
- Utiliser les clauses SET NEWNAME 20-12
- Variables de substitution pour SET NEWNAME 20-13
- Définir des paramètres pour les noms de fichier 20-14
- Démarrer l'instance en mode NOMOUNT 20-16
- Vérifier la disponibilité des sauvegardes et des fichiers de journalisation archivés 20-17
- Allouer des canaux auxiliaires 20-18

Principe de l'opération de duplication RMAN 20-19
Indiquer des options pour la commande DUPLICATE 20-21
Utiliser les options supplémentaires de la commande DUPLICATE 20-22
Utiliser EM pour cloner une base de données 20-23
Quiz 20-24
Synthèse 20-25
Présentation de l'exercice 20 : Dupliquer une base de données 20-26

Annexe A : Exercices et solutions

Annexe B : Récupération d'un tablespace jusqu'à un point dans le temps

Objectifs B-2
Récupération de tablespace jusqu'à un point dans le temps (TSPITR) :
Concepts B-3
Récupération de tablespace jusqu'à un point dans le temps (TSPITR) :
Terminologie B-4
Récupération de tablespace jusqu'à un point dans le temps : Architecture B-5
Quand utiliser l'opération TSPITR ? B-7
Préparer l'exécution de l'opération TSPITR B-8
Déterminer le point cible approprié B-9
Déterminer les tablespaces pour le jeu de blocs à récupérer B-10
Identifier les relations avec des objets situés hors du jeu de récupération B-11
Identifier les objets qui seront perdus B-12
Effectuer une opération RMAN TSPITR élémentaire B-13
Effectuer une opération TSPITR entièrement automatisée B-14
Utiliser des copies d'images pour améliorer les performances
de la récupération TSPITR B-15
Utiliser Enterprise Manager pour effectuer une opération TSPITR B-16
Traitement RMAN de l'opération TSPITR B-17
Effectuer une opération RMAN TSPITR avec une instance auxiliaire
gérée par RMAN B-19
Effectuer une opération RMAN TSPITR à l'aide de votre propre
instance auxiliaire B-20
Résolution des problèmes liés à une opération RMAN TSPITR B-21
Synthèse B-22

Annexe C : Effectuer une sauvegarde et une récupération gérées par l'utilisateur

Objectifs C-2
Types de sauvegarde et de récupération C-3
Réaliser une sauvegarde de la base de données gérée par l'utilisateur C-4
Nécessité du mode sauvegarde C-5
Identifier les fichiers à sauvegarder manuellement C-6
Sauvegarder manuellement une base de données NOARCHIVELOG C-7
Sauvegarder manuellement une base de données ARCHIVELOG C-8

Sauvegarder le fichier de contrôle C-9
Effectuer une récupération complète de la base de données gérée
par l'utilisateur : Présentation C-10
Effectuer une récupération base fermée complète : Présentation C-11
Identifier les fichiers liés à la récupération C-12
Restaurer des fichiers liés à la récupération C-13
Appliquer les données de journalisation C-15
Effectuer une récupération complète base de données ouverte C-16
Effectuer une récupération incomplète gérée par l'utilisateur : Présentation C-18
Choisir une méthode de récupération incomplète C-19
Effectuer une récupération incomplète gérée par l'utilisateur C-20
Effectuer une récupération incomplète gérée par l'utilisateur : Procédure C-22
Récupération jusqu'à un point dans le temps gérée par l'utilisateur :
Exemple C-23
Exemple de récupération jusqu'à annulation gérée par l'utilisateur C-25
Synthèse C-27

Annexe D : Gérer l'instance ASM

Objectifs D-2
Avantages d'ASM pour les administrateurs D-3
Instance ASM D-4
Composants d'une instance ASM : Processus principaux D-6
Paramètres d'initialisation d'une instance ASM D-7
Interaction entre les instances de base de données et ASM D-9
Instance ASM : Vues dynamiques des performances D-10
Privilèges système ASM D-11
Utiliser Enterprise Manager pour gérer les utilisateurs ASM D-12
Démarrer et arrêter des instances ASM à l'aide de SQL*Plus D-13
Démarrer et arrêter des instances ASM à l'aide de `srvctl` D-15
Démarrer et arrêter des instances ASM à l'aide de l'utilitaire `asmcmd` D-16
Présentation des groupes de disques D-17
Disques ASM D-18
Unités d'allocation D-19
Fichiers ASM D-20
Topographie des extents D-21
Niveau de détail du striping D-22
Striping fin D-23
Groupes d'échec ASM D-25
Striping et mise en miroir : Exemple D-26
Exemple de panne de disque D-27
Gérer les groupes de disques D-28
Créer et supprimer des groupes de disques à l'aide de SQL*Plus D-29
Ajouter des disques à des groupes D-30
Commandes `ALTER` diverses D-31

Gérer ASM à l'aide d'Enterprise Manager	D-32
Compatibilité des groupes de disques ASM	D-33
Attributs de groupe de disques ASM	D-35
Utiliser Enterprise Manager pour modifier les attributs de groupe de disques	D-36
Extraire des métadonnées ASM	D-37
Présentation de la fonctionnalité ASM Fast Mirror Resync	D-38
Synthèse	D-39

Utiliser la technologie Flashback II



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Laurent PERPETUE (laurent.perpetue@stfr.fr) has a non-transferable license to use this Student Guide

Objectifs

> Total Recall
Flashback Drop

A la fin de ce chapitre, vous pourrez :

- décrire et utiliser Oracle Total Recall
 - créer et activer une Flashback Data Archive
 - gérer et activer les Flashback Data Archives
 - afficher les métadonnées
- décrire et utiliser les corbeilles Flashback
 - restaurer des tables supprimées à partir de la corbeille
 - gérer l'utilisation de l'espace dans la corbeille
 - interroger la corbeille

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Présentation d'Oracle Total Recall

Suivi automatisé de l'historique des modifications de la base :

- Activez-le au niveau table avec la durée de conservation de votre choix.
- Les modifications qui suivent sont stockées de manière transparente et sécurisée.
- Les données pour lesquelles la durée de conservation est dépassée sont supprimées automatiquement.
- Utiliser les technologies Flashback pour interroger les données historiques.

```
SELECT ... AS OF TIMESTAMP...  
SELECT ... VERSIONS BETWEEN TIMESTAMP and TIMESTAMP...
```



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Présentation d'Oracle Total Recall

L'option Oracle Total Recall d'Oracle Database 11g (également appelée Flashback Data Archive) permet d'assurer un suivi sécurisé, efficace, simple d'utilisation et transparent des modifications apportées à une base de production.

Avec cette technologie, les données sont automatiquement stockées dans des tables activées pour Flashback Data Archive. Les interrogations flashback obtiennent ainsi un accès de niveau SQL aux versions des objets de base de données sans recevoir d'erreur Snapshot Too Old.

Une Flashback Data Archive permet le suivi et le stockage de toutes les modifications transactionnelles apportées à une table "suivie" pendant toute sa durée de vie. Il n'est plus nécessaire d'intégrer cette fonctionnalité à l'application. Vous pouvez utiliser Oracle Total Recall pour créer des états de conformité ou d'audit, analyser des données et fournir des fonctionnalités d'aide à la décision. Le processus en arrière-plan Flashback Data Archive démarre avec la base de données.

Présentation d'Oracle Total Recall (suite)

Exemples de cas d'utilisation :

- Audit : Trouver les déclarations de sinistre en double au cours de l'année précédente.
- Conformité réglementaire : Surveiller les opérations sur actions pendant une période d'embargo (quiet period).
- Gestion du cycle de vie des informations : Garantir un accès immuable à l'historique des patients.
- Stratégie de conservation : Purger automatiquement les enregistrements de plus de cinq ans.
- Etats historiques : Générer le relevé de compte d'un client.
- Récupération après erreur : Restaurer les enregistrements supprimés ou mis à jour par erreur.

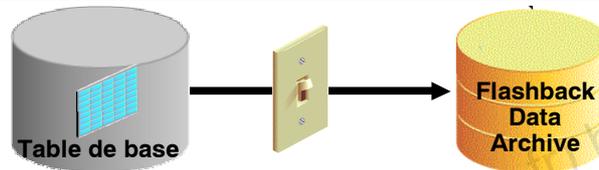
Processus de configuration

1. Créez un nouveau tablespace pour le stockage de la Flashback Data Archive.
2. Avec le privilège `FLASHBACK ARCHIVE ADMINISTER` : créez une Flashback Data Archive, affectez-la au tablespace et indiquez sa durée de conservation.

```
CREATE FLASHBACK ARCHIVE fda1  
TABLESPACE fda_tbs1 QUOTA 10M RETENTION 1 YEAR;
```

3. A l'aide du privilège objet `FLASHBACK ARCHIVE` : Modifiez les tables de base pour activer l'archivage et pour les affecter à une Flashback Data Archive.

```
ALTER TABLE HR.EMPLOYEES FLASHBACK ARCHIVE fda1;
```



ORACLE

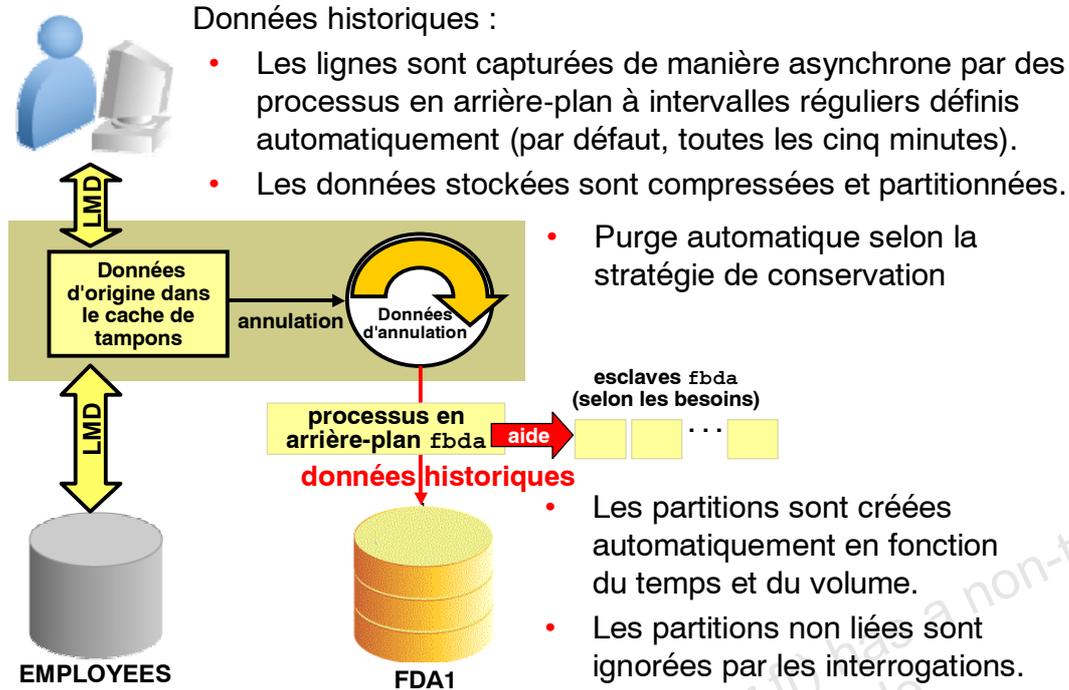
Copyright © 2009, Oracle. Tous droits réservés.

Processus de configuration

Une Flashback Data Archive comporte un ou plusieurs tablespaces. Vous pouvez disposer de plusieurs Flashback Data Archives. Celles-ci sont configurées avec une durée de conservation. Selon vos exigences en matière de durée de conservation, vous devez créer différentes Flashback Data Archives ; par exemple, une pour tous les enregistrements qui doivent être conservés pendant deux ans et une autre pour tous les enregistrements qui doivent être conservés pendant cinq ans. La base de données purgera automatiquement toutes les informations historiques le lendemain du jour où la période de conservation expire.

1. Créez un tablespace pour votre Flashback Data Archive. La taille dépend de la table de base et de l'activité LMD et LDD attendue.
2. Créez une Flashback Data Archive avec une durée de conservation. Les données qui y sont archivées sont conservées pendant ce délai. Cette tâche nécessite le privilège système `FLASHBACK ARCHIVE ADMINISTER`. Si vous devez définir plusieurs durées de conservation, créez plusieurs archives.
3. Activez l'archivage Flashback (puis désactivez-le) pour l'ensemble d'une table. Cette tâche nécessite le privilège objet `FLASHBACK ARCHIVE`. Même dans le cas où l'archivage Flashback est activé sur une table, certaines instructions LDD ne sont pas autorisées. Par défaut, il est désactivé pour toutes les tables.

Fonctionnement d'Oracle Total Recall



Copyright © 2009, Oracle. Tous droits réservés.

ORACLE

Fonctionnement d'Oracle Total Recall

Les données historiques sont capturées à partir des données d'annulation (et du cache de tampons) par le processus en arrière-plan fbda selon des intervalles réguliers définis automatiquement. L'intervalle par défaut est de cinq minutes. Une ligne mise à jour de la table de base est stockée dans son intégralité quel que soit le nombre de colonnes modifiées.

- Les données historique sont comprimées selon la compression de table OLTP et non selon la compression de colonne hybride.
Remarque : Si la table de base utilise la compression de colonne hybride, elle ne peut pas être activée pour l'archivage Flashback.
- Chaque partition de Flashback Data Archive a une durée d'au moins un jour et fait au moins 1 Mo, avec un partitionnement sur ENDSCN. Les interrogations Flashback sur les archives ignorent les partitions non liées.
- Le processus fbda peut appeler jusqu'à dix processus esclaves d'archivage Flashback.
- Si ces processus sont trop occupés, l'archivage peut être effectué en ligne, mais cela affecte notablement les temps de réponse.

Scénario utilisant Oracle Total Recall

Utilisez Flashback Data Archive pour accéder à des données historiques :

```
-- create the Flashback Data Archive
CREATE FLASHBACK ARCHIVE DEFAULT fla1
    TABLESPACE tbs1 QUOTA 10G RETENTION 5 YEAR;
```

1

```
-- Specify the default Flashback Data Archive
ALTER FLASHBACK ARCHIVE fla1 SET DEFAULT;
```

2

```
-- Enable Flashback Data Archive
ALTER TABLE inventory FLASHBACK ARCHIVE;
ALTER TABLE stock_data FLASHBACK ARCHIVE;
```

3

```
SELECT product_number, product_name, count FROM inventory AS
    OF TIMESTAMP TO_TIMESTAMP ('2007-01-01 00:00:00', 'YYYY-MM-
    DD HH24:MI:SS');
```

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Scénario utilisant Oracle Total Recall

Vous créez une Flashback Data Archive avec l'instruction `CREATE FLASHBACK ARCHIVE`.

- Vous pouvez éventuellement désigner une Flashback Data Archive par défaut pour le système.
- Vous devez fournir le nom de la Flashback Data Archive.
- Vous devez indiquer le nom du premier tablespace de la Flashback Data Archive.
- Vous pouvez identifier la quantité d'espace maximum que peut utiliser la Flashback Data Archive dans le tablespace. La valeur par défaut est Unlimited. A moins que le quota d'espace pour ce premier tablespace soit illimité, vous devez préciser une valeur. Sinon, une erreur ORA-55621 est renvoyée.
- Vous devez fournir le délai de conservation (nombre de jours pendant lequel le stockage dans la Flashback Data Archive des données concernant la table est garanti).

Dans le premier exemple de la diapositive, une Flashback Data Archive par défaut nommée `fla1` est créée. Elle utilise jusqu'à 10 Go du tablespace `tbs1`, dont les données seront conservées pendant cinq ans. Dans le deuxième exemple, la Flashback Data Archive par défaut est indiquée. Par défaut, le système ne comporte aucune Flashback Data Archive.

Deux méthodes permettent d'en définir une :

- Indiquez le nom d'une Flashback Data Archive existante dans la clause `SET DEFAULT` de l'instruction `ALTER FLASHBACK ARCHIVE`.
- Incluez `DEFAULT` dans l'instruction `CREATE FLASHBACK ARCHIVE` lorsque vous créez une Flashback Data Archive.

Dans le troisième exemple, la fonctionnalité Flashback Data Archive est activée.

Si la gestion automatique des annulations (AUM) est désactivée, vous recevez une erreur ORA-55614 lorsque vous tentez de modifier la table.

Scénario utilisant Oracle Total Recall

Ajoutez éventuellement de l'espace :

```
ALTER FLASHBACK ARCHIVE fla1  
ADD TABLESPACE tbs3 QUOTA 5G;
```

4

Modifiez éventuellement le délai de conservation :

```
ALTER FLASHBACK ARCHIVE fla1 MODIFY RETENTION 2 YEAR;
```

5

Purgez éventuellement les données :

```
ALTER FLASHBACK ARCHIVE fla1 PURGE BEFORE TIMESTAMP  
(SYSTIMESTAMP - INTERVAL '1' day);
```

6

Supprimez éventuellement une Flashback Data Archive :

```
DROP FLASHBACK ARCHIVE fla1;
```

7

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Scénario utilisant Oracle Total Recall (suite)

Afin d'activer l'archivage Flashback pour une table, incluez la clause `FLASHBACK ARCHIVE` dans l'instruction `CREATE TABLE` ou `ALTER TABLE`. Dans la clause `FLASHBACK ARCHIVE`, vous pouvez indiquer la Flashback Data Archive dans laquelle seront stockées les données historiques concernant la table. Il s'agit par défaut de la Flashback Data Archive utilisée par défaut par le système. Pour désactiver l'archivage Flashback pour une table donnée, indiquez `NO FLASHBACK ARCHIVE` dans l'instruction `ALTER TABLE`.

La dernière instruction de la diapositive précédente illustre comment extraire l'inventaire de tous les articles au début de l'année 2007. Continuons avec les exemples de la diapositive ci-dessus :

- L'exemple 4 ajoute jusqu'à 5 Go du tablespace `tbs3` à la Flashback Data Archive `fla1`.
- L'exemple 5 modifie le délai de conservation concernant la Flashback Data Archive `fla1` pour qu'il passe à deux ans.
- L'exemple 6 purge toutes les données historiques de plus d'un jour figurant dans la Flashback Data Archive `fla1`. Normalement, cette purge est effectuée automatiquement le lendemain de l'expiration du délai de conservation. Vous pouvez également l'effectuer manuellement pour un nettoyage ad hoc.

Scénario utilisant Oracle Total Recall (suite)

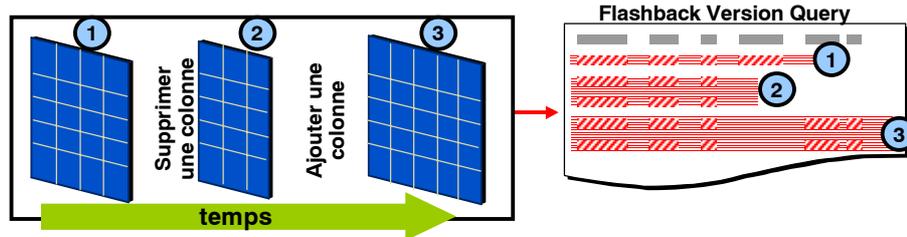
- L'exemple 7 supprime la Flashback Data Archive `fla1` ainsi que ses données historiques, mais pas ses tablespaces. Avec la commande `ALTER FLASHBACK ARCHIVE`, vous pouvez :
 - modifier le délai de conservation d'une Flashback Data Archive,
 - purger certaines de ses données ou la totalité,
 - ajouter, modifier et supprimer des tablespaces.

Remarque : La suppression de tous les tablespaces d'une Flashback Data Archive entraîne la génération d'une erreur.

Laurent PERPETUE (laurent.perpetue@srr.fr) has a non-transferable license to use this Student Guide.

Evolution transparente de schéma

- Les commandes LDD permettent les opérations suivantes :
 - Ajouter, supprimer, renommer et modifier une colonne
 - Supprimer et vider une partition
 - Renommer et vider une table



- Les interrogations Flashback Query s'appliquent à toutes les modifications LDD.
- Les autres commandes LDD *ne sont pas* automatiquement prises en charge (voir la diapositive suivante).

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Evolution transparente de schéma

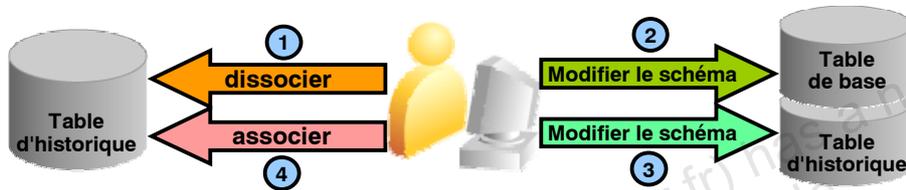
Les commandes LDD les plus courantes peuvent être utilisées avec les Flashback Data Archives. Lorsqu'un schéma évolue de l'une des façons indiquée dans la diapositive, Total Recall effectue automatiquement le suivi des modifications. Flashback Query renvoie les lignes appropriées avec le schéma correspondant (comme indiqué dans le diagramme).

Evolution complète de schéma

Dissociez ou associez des procédures à l'aide du package
DBMS_FLASHBACK_ARCHIVE :

- Désactivez Total Recall sur des tables spécifiques et autorisez des instructions LDD plus complexes (mises à niveau, fractionnement de tables, etc.).
- Assurez l'intégrité du schéma après l'association. (Les schémas de la table de base et de la table d'historique doivent être identiques.)

Remarque : Utilisez cette fonction avec précaution. Notez que la non modification de l'archive ne peut plus être garantie, car il est possible que l'historique ait été modifié pendant la période de dissociation.



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Evolution complète de schéma

Les instructions LDD qui ne sont pas automatiquement prises en charge peuvent être exécutées par le biais du package DBMS_FLASHBACK_ARCHIVE. Vous pouvez utiliser les procédures DISASSOCIATE_FBA et REASSOCIATE_FBA pour dissocier une table spécifique de sa Flashback Data Archive puis la réassocier.

Remarque : Utilisez cette fonction avec précaution. Notez que la non modification de l'archive ne peut plus être garantie, car il est possible que l'historique ait été modifié pendant la période de dissociation. Un message est consigné dans le catalogue système lorsque la dissociation se produit.

Le diagramme de la diapositive illustre le workflow suivant :

- Si vous disposez du privilège FLASHBACK_ARCHIVE_ADMINISTER, vous pouvez dissocier l'archive de la table de base.
- Apportez les modifications nécessaires à la table de base.
- Apportez les modifications nécessaires à l'archive correspondante.
- Vous pouvez alors associer la table à l'archive dans le même schéma. Total Recall vérifie que les schémas sont les mêmes à l'issue de l'association.

Restrictions

- Vous ne pouvez pas activer Total Recall pour les tables de base utilisant la compression de colonne hybride.
- Si vous effectuez une dissociation, le caractère immuable des données historiques n'est plus garanti (mais vous pouvez toujours purger ces données avec les droits appropriés).
- Les tables d'historique ne sont pas transportables.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Restrictions

Certaines instructions LDD entraînent une erreur ORA-55610 lorsqu'elles sont utilisées sur une table activée pour l'archivage Flashback. Par exemple :

- Instruction ALTER TABLE incluant une clause UPGRADE TABLE, avec ou sans clause INCLUDING DATA
- Instruction ALTER TABLE qui déplace ou échange une opération sur une partition ou une sous-partition
- Instruction DROP TABLE

Règles

- Utilisez un SCN pour des interrogations précises.
ou
- Utilisez la technologie Flashback selon vos besoins.
- Flashback utilise les paramètres système en cours.
- Pour garantir la cohérence de la base de données, exécutez toujours une opération COMMIT ou ROLLBACK avant d'interroger des données historiques.
- Vous ne pouvez pas extraire d'anciennes données à partir des vues dynamiques des performances (V\$). Elles contiennent les données actuelles.
- Toutefois, vous pouvez exécuter des interrogations sur d'anciennes données dans des vues statiques du dictionnaire telles que *_TABLES.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Règles

- Par souci de commodité, utilisez Flashback Query, Flashback Version Query ou Flashback Transaction Query pour le code SQL que vous écrivez.
- Rappelez-vous que l'ensemble du traitement Flashback utilise les paramètres de la session en cours, tels que la langue nationale et le jeu de caractères, et non les paramètres qui étaient en vigueur au moment faisant l'objet de l'interrogation.
- Pour interroger des données anciennes correspondant à un moment précis, utilisez un SCN. Si vous utilisez un horodatage, le point réel d'interrogation peut être antérieur de 3 secondes au point que vous indiquez. En effet, Oracle Database utilise des SCN en interne et les met en correspondance avec les horodatages selon une granularité de 3 secondes.
- Pour obtenir un SCN à utiliser ultérieurement avec une fonctionnalité de flashback, vous pouvez utiliser la fonction `DBMS_FLASHBACK.GET_SYSTEM_CHANGE_NUMBER`.
- Pour calculer ou extraire un point dans le temps à utiliser dans une interrogation, utilisez une valeur renvoyée par une fonction comme horodatage ou argument SCN. Par exemple, ajoutez une valeur INTERVAL à la valeur de la fonction `SYSTIMESTAMP` ou soustrayez-la.
- Pour garantir la cohérence de la base de données, exécutez toujours une opération COMMIT ou ROLLBACK avant d'interroger des données historiques.
- Vous ne pouvez pas extraire d'anciennes données à partir d'une vue dynamique des performances (V\$). En effet, une interrogation sur une vue de ce type renvoie toujours des données actuelles. Toutefois, vous pouvez exécuter des interrogations sur d'anciennes données dans des vues statiques du dictionnaire telles que *_TABLES.

Consulter des Flashback Data Archives

Consultez les résultats :

Nom de la vue (DBA/USER)	Description
*_FLASHBACK_ARCHIVE	Affiche des informations sur les Flashback Data Archives
*_FLASHBACK_ARCHIVE_TS	Affiche les tablespaces des Flashback Data Archives
*_FLASHBACK_ARCHIVE_TABLES	Affiche des informations sur les tables activées pour l'archivage Flashback

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Consulter des Flashback Data Archives

Vous pouvez utiliser les vues dynamiques du dictionnaire de données pour consulter les tables suivies et les métadonnées Flashback Data Archive. Pour accéder aux vues USER_FLASHBACK_*, vous devez disposer de privilèges de propriété sur la table concernée. Pour consulter les vues DBA_FLASHBACK_*, vous avez besoin des privilèges SYSDBA.

Pour plus d'informations, reportez-vous au manuel *Oracle Database PL/SQL Packages and Types Reference*.

Quiz

Vous ne pouvez pas supprimer une table faisant l'objet d'un suivi avec Oracle Total Recall, mais vous pouvez la vider.

1. Vrai
2. Faux

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Réponse : 1

Laurent PERPETUE (laurent.perpetue@sr.fr) has a non-transferable license to use this Student Guide

Quiz

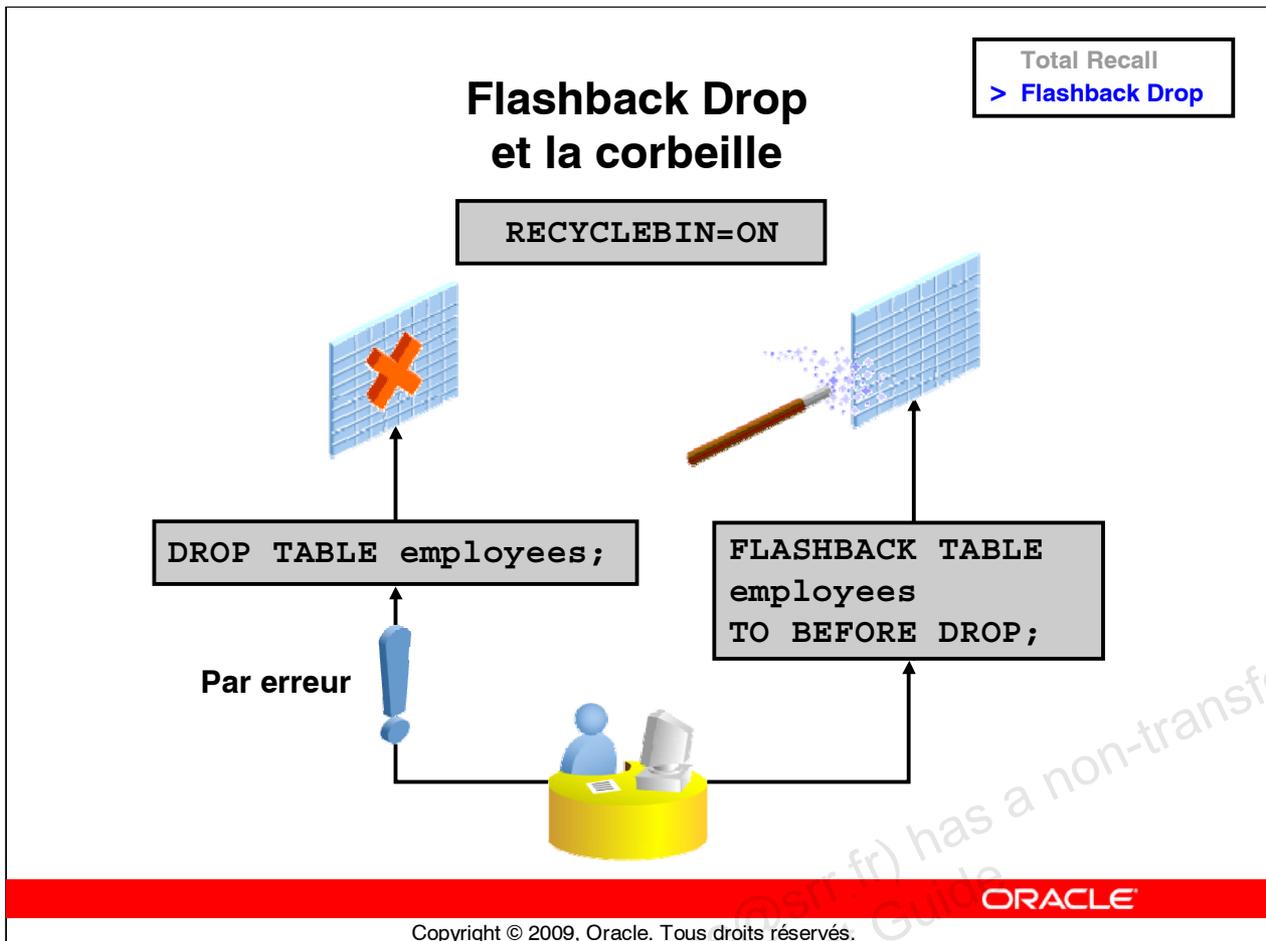
Sélectionnez toutes les affirmations qui sont vraies à propos d'Oracle Total Recall :

1. Oracle Total Recall est activé par défaut.
2. Une Flashback Data Archive permet le suivi et le stockage de toutes les modifications transactionnelles apportées à une table "suivie" pendant toute sa durée de vie.
3. La suppression d'une colonne dans une table activée pour Flashback Data Archive entraîne la génération d'une erreur.
4. Le traitement Flashback utilise toujours les paramètres qui étaient en vigueur au moment correspondant à l'interrogation.
5. Flashback utilise les paramètres de la session en cours, tels que la langue nationale et le jeu de caractères.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

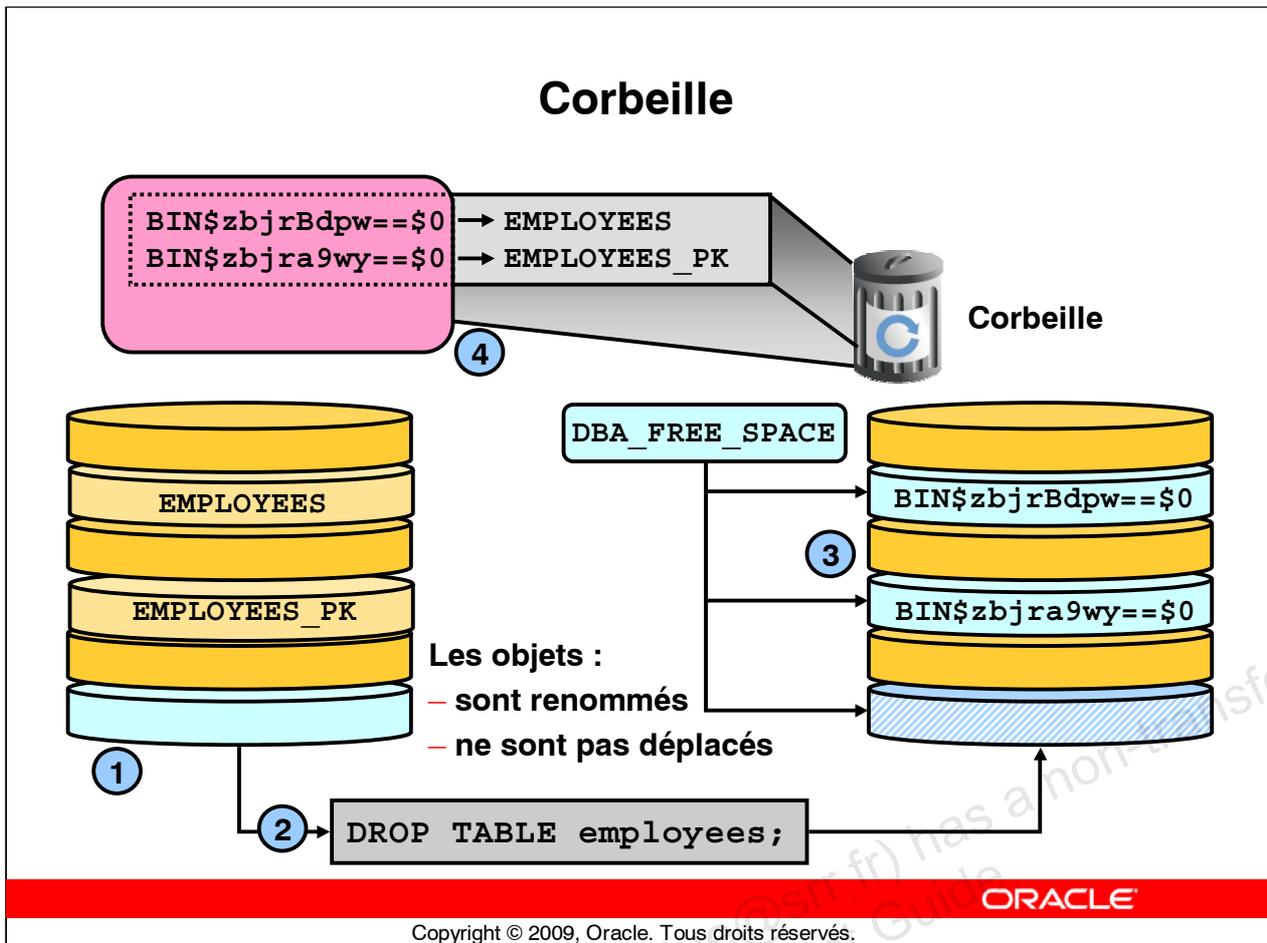
Réponses : 2, 5



Flashback Drop et la corbeille

A l'aide de la commande `FLASHBACK TABLE`, vous pouvez annuler les effets d'une instruction `DROP TABLE` sans avoir recours à la récupération jusqu'à un point dans le temps.

Remarque : Le paramètre d'initialisation `RECYCLEBIN` permet de contrôler si la fonction Flashback Drop est activée (`ON`) ou désactivée (`OFF`). Lorsqu'il a la valeur `OFF`, les tables supprimées ne sont pas placées dans la corbeille. Lorsqu'il a la valeur `ON`, les tables supprimées sont placées dans la corbeille et peuvent être récupérées. Par défaut, le paramètre `RECYCLEBIN` a la valeur `ON`.



Corbeille

Si vous supprimez une table alors que la corbeille n'est pas activée, l'espace associé à cette table et aux objets dépendants est immédiatement récupérable (c'est-à-dire qu'il peut être utilisé pour d'autres objets).

En revanche, si vous supprimez une table alors que la corbeille est activée, l'espace associé à la table et aux objets dépendants n'est pas immédiatement récupérable, même s'il apparaît dans DBA_FREE_SPACE. Les objets supprimés sont référencés dans la corbeille et appartiennent toujours à leur propriétaire. L'espace utilisé par les objets de la corbeille n'est jamais récupéré automatiquement, sauf en cas de manque d'espace. Cela vous permet de récupérer les objets de la corbeille pendant la durée la plus importante possible.

Lorsqu'une table supprimée est "déplacée" vers la corbeille, elle est renommée à l'aide d'un nom généré par le système, de même que les objets et contraintes associés. La convention d'attribution d'un nouveau nom est la suivante :

BIN\$unique_id\$version

où `unique_id` est un identificateur unique de niveau global (GUID) de 26 caractères pour cet objet (rendant ainsi le nom de la corbeille unique sur l'ensemble des bases de données) et `version` est un numéro de version attribué par la base.

Corbeille (suite)

La corbeille proprement dite est une table du dictionnaire de données qui gère les relations entre les noms originaux des objets supprimés et les noms générés par le système.

Vous pouvez interroger la corbeille à l'aide de la vue `DBA_RECYCLEBIN`. Le diagramme de la diapositive illustre ce comportement :

1. Vous avez créé une table nommée `EMPLOYEES` dans le tablespace.
2. Vous supprimez la table `EMPLOYEES`.
3. Les extents (ensembles de blocs contigus) occupés par la table `EMPLOYEES` sont à présent considérés comme de l'espace libre.
4. La table `EMPLOYEES` est renommée et son nouveau nom est enregistré dans la corbeille.

Restaurer des tables à partir de la corbeille

- Restaurer les tables supprimées et les objets dépendants.
- Si plusieurs entrées de la corbeille portent le même nom d'origine :
 - Utilisez les noms uniques générés par le système pour restaurer une version particulière.
 - Lors de l'utilisation des noms d'origine, le système applique le principe LIFO pour déterminer la table à restaurer.
- Modifiez le nom d'origine s'il est déjà utilisé.

```
FLASHBACK TABLE <table_name> TO BEFORE DROP
[RENAME TO <new_name>];
```

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Restaurer des tables à partir de la corbeille

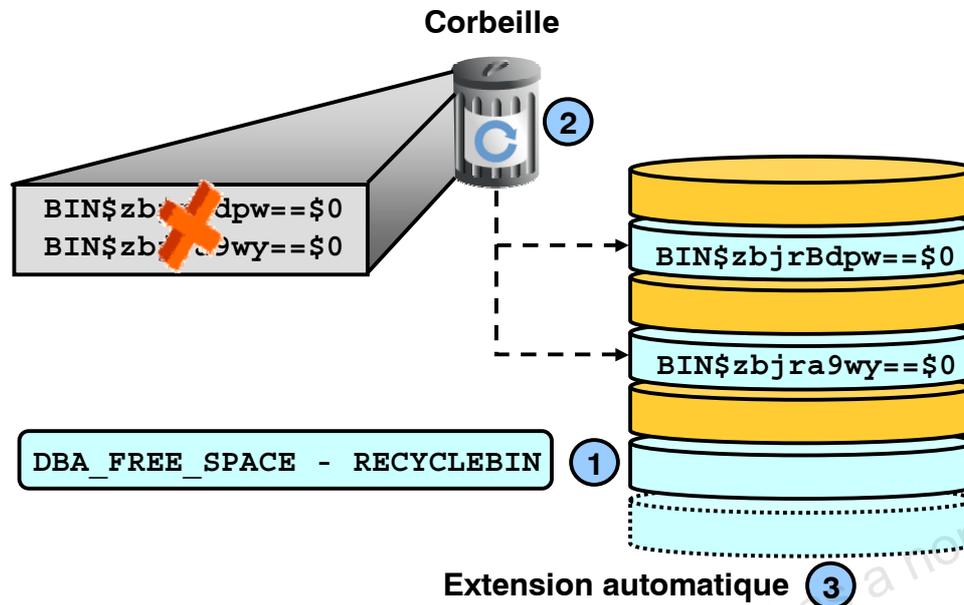
Utilisez la commande `FLASHBACK TABLE . . . TO BEFORE DROP` pour récupérer une table et tous les objets dépendants à partir de la corbeille. Vous pouvez indiquer soit le nom d'origine de la table, soit le nom généré par le système affecté à l'objet lors de sa suppression.

Si vous indiquez le nom d'origine et que la corbeille contient plusieurs objets portant ce nom, l'objet qui a été déplacé le plus récemment vers la corbeille est récupéré en premier (LIFO : Last In, First Out - dernier entré, premier sorti). Si vous souhaitez extraire une version plus ancienne de la table, vous pouvez indiquer le nom généré par le système de la table que vous souhaitez extraire, ou exécuter des instructions `FLASHBACK TABLE TO BEFORE DROP` jusqu'à ce que vous obteniez la table voulue.

Si une nouvelle table portant le même nom a été créée dans le même schéma depuis la suppression de la table d'origine, une erreur est renvoyée, sauf si vous indiquez également la clause `RENAME TO`.

Remarque : Lorsque vous procédez à un flashback d'une table supprimée, les index, les déclencheurs (triggers) et les contraintes récupérés conservent le nom qu'ils portaient dans la corbeille. Par conséquent, il est conseillé d'interroger la corbeille et la vue `DBA_CONSTRAINTS` avant de procéder au flashback d'une table supprimée. De cette façon, vous pouvez renommer les index, les déclencheurs et les contraintes récupérés avec des noms plus conviviaux.

Corbeille : Récupération automatique d'espace



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Corbeille : Récupération automatique d'espace

Tant que l'espace utilisé par des objets de la corbeille n'est pas réutilisé, vous pouvez récupérer ces objets à l'aide de la fonctionnalité Flashback Drop. Les stratégies de récupération de l'espace utilisé par les objets de la corbeille sont les suivantes :

- Nettoyage manuel lorsque vous exécutez explicitement une commande PURGE.
- Nettoyage automatique en cas de manque d'espace : alors que les objets se trouvent dans la corbeille, l'espace correspondant est également signalé dans DBA_FREE_SPACE, car il peut être récupéré automatiquement. L'espace libre dans un tablespace particulier est ensuite consommé dans l'ordre suivant :
 1. Espace libre ne correspondant pas à des objets de la corbeille.
 2. Espace libre correspondant à des objets de la corbeille. Dans ce cas, les objets de la corbeille sont automatiquement purgés de la corbeille via un algorithme FIFO (First In, First Out - premier entré, premier sorti).
 3. De l'espace libre est alloué automatiquement si le tablespace est en auto-extension. Supposons que vous définissiez une nouvelle table dans le tablespace TBS1. Si de l'espace libre alloué à ce tablespace ne correspond pas à un objet de la corbeille, cet espace libre est utilisé comme première étape. Si cela ne suffit pas, l'espace libre correspondant aux objets de la corbeille résidant dans TBS1 est utilisé. Si l'espace libre de certains objets de la corbeille est utilisé, ces objets sont éliminés automatiquement de la corbeille. A ce stade, vous ne pouvez plus récupérer ces objets à l'aide de la fonctionnalité Flashback Drop. En dernier recours, si les besoins en termes d'espace ne sont pas encore satisfaits, le tablespace TBS1 est étendu, dans la mesure du possible.

Corbeille : Récupération manuelle d'espace

```
PURGE {TABLE <table_name> | INDEX <index_name>}
```

```
PURGE TABLESPACE <ts_name> [USER <user_name>]
```

```
PURGE [USER_ | DBA_] RECYCLEBIN
```

ORACLE Enterprise Manager 11g Database Control

Database Instance: orcl > Tables > Recycle Bin

When you drop a table from a non-system, locally managed tablespace, Oracle does not immediately reclaim the space associated with the table. Oracle places the table and any associated objects in the Recycle Bin, where, in case the table was dropped in error, it can be recovered (Flashback Drop) at a later time.

Search

Schema Name: HR Table: [] Go

Results

Purge Flashback Drop

Select	Object Name	Schema	Recovery Scope	Tablespace	Drop Time	Create Time	Size	Operation
<input type="checkbox"/>	Recycle Bin							View Content
<input checked="" type="checkbox"/>	EMPLOYEES2	HR	TABLE	USERS	2007-07-02:15:45:13	2007-07-02:15:44:50.8		View Content

Copyright © 2009, Oracle. Tous droits réservés.

Corbeille : Récupération manuelle d'espace

Utilisez la commande PURGE pour supprimer de manière permanente des objets de la corbeille. Lorsqu'un objet est purgé de la corbeille, il est supprimé de façon permanente de la base de données, de même que les objets dépendants. Ces objets ne peuvent plus être récupérés à l'aide de la fonctionnalité Flashback Drop. Voici quelques exemples d'utilisation de la commande PURGE :

- PURGE TABLE purge la table désignée.
- PURGE INDEX purge l'index désigné.
- PURGE TABLESPACE purge tous les objets qui résident dans le tablespace désigné. En outre, les objets qui résident dans d'autres tablespaces peuvent être purgés s'ils sont dépendants.
- PURGE RECYCLEBIN purge tous les objets appartenant à l'utilisateur en cours. RECYCLEBIN et USER_RECYCLEBIN sont synonymes.
- PURGE DBA_RECYCLEBIN purge tous les objets. Vous devez disposer de privilèges système suffisants, ou du privilège système SYSDBA, pour exécuter cette commande.

Les tables peuvent également être purgées de la corbeille à l'aide d'Enterprise Manager. Dans l'onglet Schema, cliquez sur Tables, puis sélectionnez le schéma dans lequel résidait l'objet supprimé et cliquez sur le bouton Recycle Bin. Sélectionnez la table dans la liste des résultats et cliquez sur le bouton Purge.

Remarque : S'agissant des commandes PURGE TABLE et PURGE INDEX, si vous indiquez un nom d'origine et que la corbeille contient plusieurs objets portant ce nom, l'objet qui réside dans la corbeille depuis le plus longtemps est purgé en premier (FIFO : First In, First Out - premier entré, premier sorti).

Contourner la corbeille

```
DROP TABLE <table_name> [PURGE] ;
```

```
DROP TABLESPACE <ts_name>  
[INCLUDING CONTENTS] ;
```

```
DROP USER <user_name> [CASCADE] ;
```

Désactiver l'utilisation de la corbeille :

```
ALTER SYSTEM SET RECYCLEBIN=OFF SCOPE=SPFILE;
```



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Contourner la corbeille

Vous pouvez utiliser la commande `DROP TABLE PURGE` pour supprimer de manière permanente de la base de données une table et les objets dépendants. Lorsque vous utilisez cette commande, les objets correspondants ne sont pas déplacés vers la corbeille. Cette commande offre les mêmes fonctionnalités que la commande `DROP TABLE` des versions précédentes.

Lorsque vous exécutez `DROP TABLESPACE . . . INCLUDING CONTENTS`, les objets du tablespace ne sont pas placés dans la corbeille. En outre, les objets de la corbeille appartenant au tablespace sont purgés. Lorsque vous exécutez la même commande sans la clause `INCLUDING CONTENTS`, le tablespace doit être vide pour que la commande aboutisse. Cependant, la corbeille peut contenir des objets appartenant au tablespace. Dans ce cas, ces objets sont purgés.

Lorsque vous exécutez la commande `DROP USER . . . CASCADE`, l'utilisateur et tous les objets appartenant à l'utilisateur sont supprimés de façon permanente de la base de données. Tous les objets de la corbeille appartenant à l'utilisateur supprimé sont purgés.

Pour accroître la sécurité, vous pouvez décider de ne pas utiliser la corbeille. Connecté en tant que `SYSDBA`, vous pouvez effectuer les opérations suivantes :

- Afficher le statut de la corbeille avec :
`SHOW PARAMETER RECYCLEBIN`
- Désactiver l'utilisation de la corbeille avec :
`ALTER SYSTEM SET RECYCLEBIN=OFF SCOPE=SPFILE;`

Après l'exécution de cette commande, vous devez redémarrer la base.

Interroger la corbeille

```
SELECT owner, original_name, object_name,  
       type, ts_name, droptime, related, space  
FROM dba_recyclebin  
WHERE can_undrop = 'YES';
```

```
SQL> SELECT original_name, object_name, ts_name, droptime  
FROM user_recyclebin WHERE can_undrop = 'YES';
```

ORIGINAL_NAME	OBJECT_NAME	TS_NAME	DROPTIME
EMPLOYEES2	BIN\$NE4Rk64w...gbpQ==	USERS	2007-07-02:15:45:13

```
SQL> SHOW RECYCLEBIN
```

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Interroger la corbeille

Vous pouvez afficher tous les objets que vous avez supprimés en interrogeant `user_recyclebin` ou `RECYCLEBIN`. `RECYCLEBIN` est un synonyme simplifiant l'utilisation.

La vue `dba_recyclebin` affiche tous les objets qui ont été supprimés par tous les utilisateurs, et qui se trouvent toujours dans la corbeille.

Vous pouvez également utiliser la commande `SQL*Plus SHOW RECYCLEBIN`.

Cette commande affiche uniquement les objets dont la suppression peut être "annulée".

Les exemples de la diapositive montrent comment extraire des informations importantes de la corbeille :

- `original_name` est le nom de l'objet avant sa suppression.
- `object_name` est le nom généré par le système de l'objet après sa suppression.
- `type` est le type de l'objet.
- `ts_name` est le nom du tablespace auquel l'objet appartient.
- `droptime` est la date à laquelle l'objet a été supprimé.
- `related` est l'identificateur de l'objet supprimé.
- `space` est le nombre de blocs actuellement utilisés par l'objet.

Vous pouvez également voir le contenu de la corbeille en utilisant Database Control.

Remarque : Pour obtenir des informations détaillées sur la vue `DBA_RECYCLEBIN`, reportez-vous au guide *Oracle Database Reference*.

Quiz

Lorsque vous procédez à un flashback d'une table supprimée, les index, les déclencheurs (triggers) et les contraintes récupérés conservent le nom qu'ils portaient dans la corbeille.

1. Vrai
2. Faux

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Réponse : 1

Synthèse

Ce chapitre vous a permis d'apprendre à :

- configurer et utiliser Total Recall
- restaurer des tables supprimées à partir de la corbeille
- interroger la corbeille

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Présentation de l'exercice 11 : Utiliser la technologie Flashback

Cet exercice porte sur les points suivants :

- Utiliser Total Recall
- Recycler des activités de la corbeille (*facultatif*)

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Unauthorized reproduction or distribution prohibited. Copyright© 2014, Oracle and/or its affiliates.

Laurent PERPETUE (laurent.perpetue@srr.fr) has a non-transferable license to use this Student Guide.

12

Utiliser Flashback Database

Copyright © 2009, Oracle. Tous droits réservés.

ORACLE

Laurent PERPETUE (laurent.perpetue@stfr.fr) has a non-transferable license to use this Student Guide

Objectifs

A la fin de ce chapitre, vous pourrez :

- configurer Flashback Database
- exécuter des opérations Flashback Database
- surveiller Flashback Database

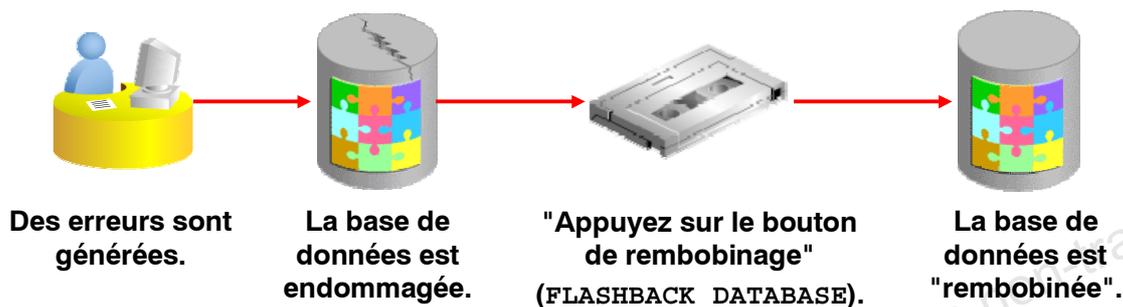
ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Flashback Database

L'opération Flashback Database :

- est semblable à une touche permettant de "rembobiner" la base de données
- peut être utilisée dans les cas de corruption logique de données par les utilisateurs



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Flashback Database

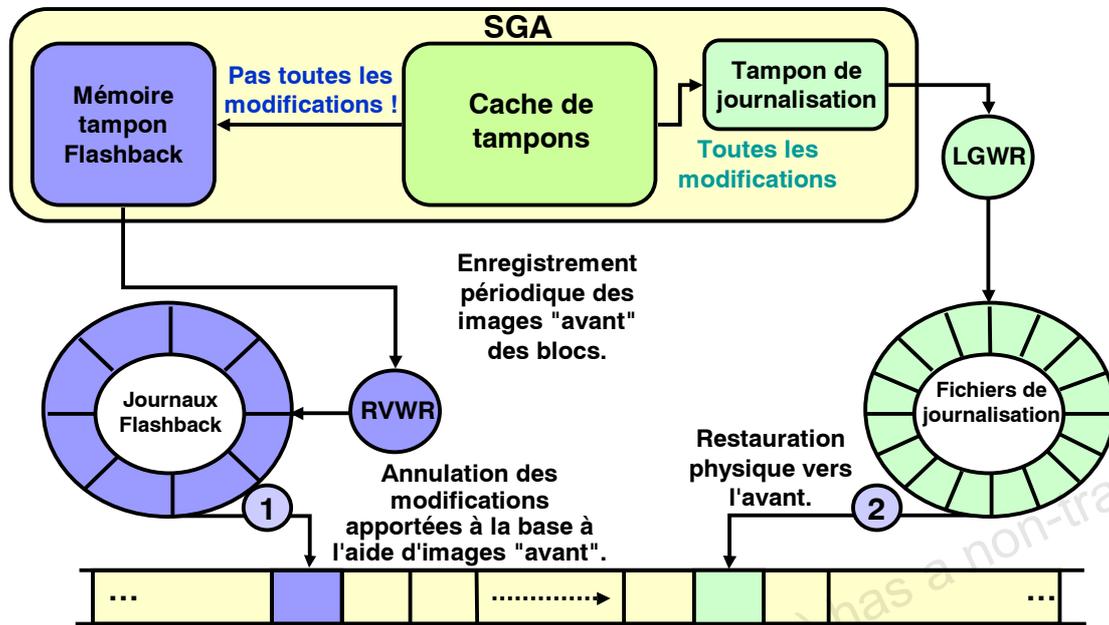
Avec Flashback Database, vous pouvez rétablir rapidement la base de données à un point antérieur dans le temps, en annulant toutes les modifications apportées depuis cet instant. Cette opération est rapide, car elle ne nécessite pas la restauration de sauvegardes.

Vous pouvez utiliser cette fonctionnalité pour annuler des modifications ayant entraîné des corruptions logiques de données.

Lorsque vous utilisez Flashback Database, Oracle Database utilise d'anciennes images de bloc pour annuler (back out) les modifications apportées à la base de données. Au cours du fonctionnement normal de la base de données, Oracle Database consigne occasionnellement ces images de bloc dans des journaux Flashback. Ces derniers sont écrits de façon séquentielle et ne sont pas archivés. Oracle Database crée, supprime et redimensionne automatiquement les journaux Flashback dans la zone de récupération rapide. Vous n'avez à utiliser les journaux Flashback que pour surveiller les performances et déterminer l'espace disque à leur allouer dans la zone de récupération rapide.

La durée nécessaire au rembobinage d'une base de données avec Flashback Database est proportionnelle au point jusqu'auquel vous souhaitez remonter et à la quantité d'activités de base de données survenues après ce point. La durée nécessaire à la restauration et à la récupération de la base de données totale peut s'avérer bien plus longue. Les images "avant" figurant dans les journaux Flashback sont utilisées uniquement pour restaurer la base jusqu'à un point dans le temps tandis que la récupération avant (forward recovery) permet de ramener la base à un état cohérent du passé. Oracle Database ramène les fichiers de données vers le point antérieur indiqué, mais pas les fichiers auxiliaires tels que les fichiers de paramètres d'initialisation. La fonctionnalité Flashback Database peut également être utilisée en complément de Data Guard et Recovery Advisor, ou pour synchroniser des bases de données dupliquées.

Architecture Flashback Database



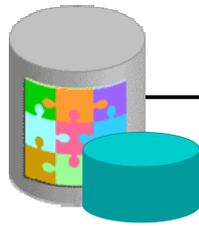
Copyright © 2009, Oracle. Tous droits réservés.

Architecture Flashback Database

Lorsque vous activez la fonctionnalité Flashback Database, le processus en arrière-plan RVWR (Flashback Writer) est démarré. Ce processus écrit de manière séquentielle les données Flashback Database à partir de la mémoire tampon Flashback vers les journaux Flashback Database, lesquels sont réutilisés de manière circulaire. Par la suite, lorsqu'une commande `FLASHBACK DATABASE` est exécutée, les journaux Flashback sont utilisés pour la restauration des images "avant" des blocs, puis les données de journalisation sont utilisées pour la réimplémentation des modifications jusqu'à l'heure de flashback souhaitée.

La surcharge liée à l'activation de Flashback Database dépend du mélange lecture/écriture de la charge globale de la base de données. Etant donné que les interrogations n'ont pas besoin de consigner de données flashback, plus la charge globale implique un nombre élevé d'opérations d'écriture, plus la surcharge liée à l'activation de Flashback Database est élevée.

Configurer Flashback Database



1. Configurez la zone de récupération rapide.



2. Définissez le délai de conservation souhaité.



3. Activez Flashback Database.

```
SQL> SHUTDOWN IMMEDIATE
```

```
SQL> STARTUP MOUNT
```

```
SQL> ALTER DATABASE ARCHIVELOG;
```

```
SQL> ALTER SYSTEM SET
```

```
2 DB_FLASHBACK_RETENTION_TARGET=2880 SCOPE=BOTH;
```

```
SQL> ALTER DATABASE FLASHBACK ON;
```

```
SQL> ALTER DATABASE OPEN;
```

Si votre base de données est en mode ARCHIVELOG, il n'y a pas besoin de la redémarrer.

Avec la base de données ouverte

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Configurer Flashback Database

Vous pouvez configurer Flashback Database comme suit :

1. Configurez la zone de récupération rapide.
2. Définissez le délai de conservation souhaité à l'aide du paramètre d'initialisation `DB_FLASHBACK_RETENTION_TARGET`. Vous pouvez indiquer une limite supérieure, en minutes, concernant l'instant jusqu'auquel vous souhaitez pouvoir procéder au flashback de la base. L'exemple de la diapositive indique 2 880 minutes, ce qui représente deux jours. Ce paramètre n'est qu'un objectif et n'offre aucune garantie. L'intervalle de flashback dépend de la quantité de données flashback conservées dans la zone de récupération rapide.
3. Activez Flashback Database à l'aide de la commande suivante :

```
ALTER DATABASE FLASHBACK ON;
```

Pour que vous puissiez exécuter la commande permettant d'activer Flashback Database, la base doit être configurée pour l'archivage.

Vous pouvez utiliser l'interrogation suivante pour déterminer si Flashback Database est activé :

```
SELECT flashback_on FROM v$database;
```

Vous pouvez désactiver Flashback Database avec la commande `ALTER DATABASE FLASHBACK OFF`. Par conséquent, tous les journaux Flashback Database existants sont supprimés automatiquement.

Remarque : Vous ne pouvez activer Flashback Database que lorsque la base de données est montée en mode exclusif, et non ouverte.

Opérations à effectuer

Workflow de configuration :

1. Assurez-vous que la base de données est en mode ARCHIVELOG.
2. Activez la journalisation flashback et désignez une zone de récupération rapide.

Flash Recovery

This database is using a flash recovery area. The chart shows space used by each file type that is not reclaimable by Oracle. Performing backups to tertiary storage is one way to make space reclaimable. Usable Flash Recovery Area includes free and reclaimable space.

Flash Recovery Area Location

Flash Recovery Area Size GB

Flash Recovery Area Size must be set when the location is set.

Non-reclaimable Flash Recovery Area (GB)	2.07
Reclaimable Flash Recovery Area (GB)	1.38
Free Flash Recovery Area (GB)	6.55

Enable Flashback Database*

Flashback database can be used for fast database point-in-time recovery, as it returns the database to a prior point-in-time without restoring files. Flashback is the preferred point-in-time recovery method in the recovery wizard when appropriate. The flash recovery area must be set to enable flashback database.

Flashback Retention Time Hours

Current size of the flashback logs(GB) n/a

Lowest SCN in the flashback data n/a

Flashback Time n/a

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Opérations à effectuer

Connectez-vous à Enterprise Manager (EM). Dans la page Availability, sélectionnez Recovery Settings dans la région Backup/Recovery. Assurez-vous que la base de données est en mode ARCHIVELOG. Si tel n'est pas le cas, cochez la case ARCHIVELOG Mode, puis cliquez sur Continue. Vous devez arrêter et redémarrer l'instance pour que les modifications prennent effet.

Lorsque la zone de récupération rapide et l'archivage sont configurés, la valeur 10 est affectée au paramètre `USE_DB_RECOVERY_FILE_DEST` pour la destination des fichiers de journalisation archivés. Activez la journalisation flashback en cochant la case "Enable Flashback Database - flashback logging can be used for fast database point-in-time recovery". Vous pouvez également définir le délai de conservation flashback et afficher des informations importantes concernant la fenêtre Flashback Database.

Examinez l'emplacement de la zone de récupération rapide. La zone de récupération rapide est un emplacement de stockage unifié pour l'ensemble des fichiers et activités liés à la récupération dans une base de données Oracle. Elle regroupe tous les fichiers nécessaires pour récupérer intégralement une base suite à une défaillance physique. Les fichiers liés à la récupération qui peuvent être créés dans la zone de récupération rapide sont les suivants : les fichiers de journalisation archivés, les fichiers de contrôle, les sauvegardes créées par Recovery Manager (RMAN), les journaux Flashback et le fichier de suivi des modifications. En allouant un emplacement de stockage et en unifiant les fichiers liés à la récupération dans une zone spécifique, le serveur de base de données Oracle évite à l'administrateur d'avoir à gérer les fichiers disque créés par ces composants. L'emplacement par défaut de la zone de récupération rapide est `$ORACLE_BASE/flash_recovery_area`. Si vous souhaitez utiliser un emplacement différent, changez-le maintenant. Faites défiler la page Recovery Settings jusqu'en bas et cliquez sur Apply.

Flashback Database : Exemples

- Pour effectuer un flash back : base de données montée (en mode exclusif)

```
RMAN> FLASHBACK DATABASE TO TIME =
  2> "TO_DATE('2009-05-27 16:00:00',
  3> 'YYYY-MM-DD HH24:MI:SS')";

RMAN> FLASHBACK DATABASE TO SCN=23565;
RMAN> FLASHBACK DATABASE
  2> TO SEQUENCE=223 THREAD=1;

SQL> FLASHBACK DATABASE
  2 TO TIMESTAMP(SYSDATE-1/24);
SQL> FLASHBACK DATABASE TO SCN 53943;
SQL> FLASHBACK DATABASE TO RESTORE POINT b4_load;
```

Surveillez la progression de Flashback Database à l'aide de la vue V\$SESSION_LONGOPS.

- Pour vérifier les modifications : base de données ouverte en mode lecture seule
- Pour finaliser : base de données ouverte en mode lecture/écriture avec l'option RESETLOGS

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Flashback Database : Exemples

Vous pouvez utiliser la commande RMAN FLASHBACK DATABASE afin d'exécuter l'opération Flashback Database. Vous pouvez utiliser SEQUENCE et THREAD pour indiquer un numéro de séquence et un thread de fichier de journalisation comme limite inférieure. RMAN sélectionne uniquement les fichiers qui peuvent être utilisés pour effectuer le flashback jusqu'au numéro de séquence indiqué (non inclus).

Vous pouvez également utiliser la commande SQL FLASHBACK DATABASE pour rétablir la base de données à un instant ou un numéro SCN passé. Si vous utilisez la clause TO SCN, vous devez indiquer un numéro. Si vous utilisez TO TIMESTAMP, vous devez indiquer un horodatage. Vous pouvez également indiquer un nom de point de restauration.

Vous pouvez surveiller la progression de Flashback Database à l'aide de la vue V\$SESSION_LONGOPS.

Remarque : La base de données doit être montée en mode exclusif pour l'exécution de la commande FLASHBACK DATABASE et ouverte en lecture seule pour l'examen des modifications. Elle doit ensuite être ouverte en lecture/écriture avec l'option RESETLOGS une fois l'opération terminée.

Considérations relatives à Flashback Database

- Une fois l'opération Flashback Database terminée, ouvrez la base de données :
 - En mode lecture seule afin de vérifier que l'heure ou le numéro SCN cible approprié a été utilisé
 - Avec l'option `RESETLOGS` pour autoriser les opérations LMD
- Le contraire d'une opération "flashback" est une opération de "récupération".
- Vous ne pouvez pas utiliser Flashback Database dans les situations suivantes :
 - Le fichier de contrôle a été restauré ou recréé.
 - Un tablespace a été supprimé.
 - La taille d'un fichier de données a été réduite.
- Utilisez la clause `TO BEFORE RESETLOGS` pour procéder à un flashback jusqu'au point situé avant la dernière opération `RESETLOGS`.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Considérations relatives à Flashback Database

Lorsque vous ne pouvez pas utiliser la fonctionnalité Flashback Database, vous devez utiliser une opération de récupération incomplète afin de rétablir la base de données dans un état antérieur. Une fois l'opération Flashback Database terminée, vous pouvez ouvrir la base en mode lecture seule afin de vérifier que l'heure ou le numéro SCN cible approprié a été utilisé. A défaut, vous pouvez procéder à un nouveau flashback de la base ou effectuer une récupération afin de réimplémenter les modifications. Par conséquent, pour annuler une opération Flashback Database, vous devez récupérer la base de données vers l'avant.

Vous ne pouvez pas utiliser Flashback Database pour récupérer un fichier de données qui a été supprimé au cours de l'intervalle sur lequel vous procédez au flashback. Le fichier de données supprimé est ajouté au fichier de contrôle et marqué comme étant hors ligne (offline), mais il n'est pas soumis au flashback. Flashback Database ne peut pas procéder au flashback d'un fichier de données jusqu'à une heure postérieure à sa création et avant l'opération de redimensionnement. Si un fichier a été redimensionné au cours de l'intervalle vers lequel vous allez procéder au flashback de la base de données, vous devez mettre le fichier hors ligne avant de commencer l'opération Flashback Database. Cette règle s'applique aux fichiers qui ont fait l'objet d'une récupération d'espace plutôt que d'une extension. Vous pouvez utiliser Flashback Database avec les fichiers de données configurés pour l'extension automatique. Vous pouvez procéder à un flashback jusqu'au point situé juste avant la dernière opération `RESETLOGS` en indiquant la clause `TO BEFORE RESETLOGS` dans la commande `FLASHBACK DATABASE`.

Remarque : Le délai de conservation flashback cible (paramètre `DB_FLASHBACK_RETENTION_TARGET`) ne garantit pas de manière absolue la disponibilité des données flashback. Si de l'espace est nécessaire pour des fichiers requis dans la zone de récupération, les journaux flashback peuvent être supprimés automatiquement.

Surveiller Flashback Database

Pour surveiller la capacité à atteindre le délai de conservation cible :

- Affichez l'espace disque requis pour la zone de récupération rapide :

```
SQL> SELECT estimated_flashback_size,
2         flashback_size
3 FROM    V$FLASHBACK_DATABASE_LOG;
```

- Déterminez la fenêtre flashback actuelle :

```
SQL> SELECT oldest_flashback_scn,
2         oldest_flashback_time
3 FROM    V$FLASHBACK_DATABASE_LOG;
```

- Surveillez la journalisation dans les journaux Flashback Database :

```
SQL> SELECT *
2 FROM    V$FLASHBACK_DATABASE_STAT;
```

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Surveiller Flashback Database

Il est important de surveiller l'utilisation de l'espace dans la zone de récupération rapide pour savoir si le délai de conservation est respecté. Utilisez la vue V\$FLASHBACK_DATABASE_LOG pour surveiller le délai de conservation souhaité pour Flashback Database :

- ESTIMATED_FLASHBACK_SIZE utilise les données flashback enregistrées précédemment pour évaluer la quantité d'espace disque nécessaire dans la zone de récupération rapide pour que les journaux Flashback respectent le délai de conservation flashback souhaité. Cette estimation est basée sur la charge globale depuis le démarrage de l'instance ou au cours de l'intervalle le plus récent égal au délai de conservation flashback souhaité (le plus court des deux).
- FLASHBACK_SIZE indique la taille actuelle, en octets, des données flashback.
- OLDEST_FLASHBACK_SCN et OLDEST_FLASHBACK_TIME indiquent le SCN et l'heure approximatifs jusqu'auxquels vous pouvez procéder à un flashback de la base de données. La valeur CURRENT_SCN de la vue V\$DATABASE indique le SCN actuel de la base.

Surveiller Flashback Database (suite)

Utilisez la vue V\$FLASHBACK_DATABASE_STAT pour surveiller la surcharge liée à la journalisation des données flashback dans les journaux Flashback Database. Cette vue contient 24 heures d'informations, chaque ligne représentant un intervalle d'une heure. Vous pouvez utiliser cette vue pour déterminer les changements de fréquence dans la génération des données flashback.

```
SQL> SELECT begin_time, end_time, flashback_data,
           db_data,
           2 redo_data, estimated_flashback_size AS EST_FB_SZE
           3 FROM V$FLASHBACK_DATABASE_STAT;
```

BEGIN_TIM	END_TIME	FLASHBACK_DATA	DB_DATA	REDO_DATA	EST_FB_SZE
12-FEB-09	12-FEB-09	16384	0	24576	0
12-FEB-09	12-FEB-09	6594560	7471104	1533440	815923200
12-FEB-09	12-FEB-09	17235968	12361728	5150920	839467008
12-FEB-09	12-FEB-09	311648256	37249024	10272768	855195648

A partir de ces informations, vous pouvez être amené à ajuster le délai de conservation ou la taille de la zone de récupération rapide.

FLASHBACK_DATA et REDO_DATA représentent respectivement le nombre d'octets de données flashback et de données de journalisation écrits au cours de l'intervalle. DB_DATA indique le nombre d'octets de blocs de données lus et écrits. Cette vue contient également l'espace flashback estimé nécessaire pour l'intervalle.

Vous pouvez interroger la vue V\$RECOVERY_FILE_DEST pour afficher les informations relatives à la zone de récupération rapide. Les colonnes sont les suivantes :

- **NAME** : Nom de la zone de récupération rapide, indiquant la chaîne d'emplacement.
- **SPACE_LIMIT** : Limite de disque indiquée dans le paramètre DB_RECOVERY_FILE_DEST_SIZE.
- **SPACE_USED** : Espace utilisé par les fichiers de la zone de récupération rapide (en octets).
- **SPACE_RECLAIMABLE** : Quantité d'espace pouvant être récupérée par la suppression des fichiers obsolètes, des fichiers redondants et des autres fichiers de faible priorité, par l'intermédiaire de l'algorithme de gestion de l'espace.
- **NUMBER_OF_FILES** : Nombre de fichiers.

```
SQL> SELECT name, space_limit AS quota,
           2 space_used AS used,
           3 space_reclaimable AS reclaimable,
           4 number_of_files AS files
           5 FROM v$recovery_file_dest ;
```

NAME	QUOTA	USED	RECLAIMABLE	FILES
/u01/flash_recovery_area	5368707120	2507809104	203386880	226

Surveiller Flashback Database avec EM

Flash Recovery

This database is using a flash recovery area. The chart shows space used by each file type that is not reclaimable by Oracle. Performing backups to tertiary storage is one way to make space reclaimable. Usable Flash Recovery Area includes free and reclaimable space.

Flash Recovery Area Location:

Flash Recovery Area Size: GB
Flash Recovery Area Size must be set when the location is set.

Non-reclaimable Flash Recovery Area (MB): 294

Reclaimable Flash Recovery Area (B): 0

Free Flash Recovery Area (GB): 5.71

Enable Flashback Database*
Flashback database can be used for fast database point-in-time recovery, as it returns the database to a prior point-in-time without restoring files. Flashback is the preferred point-in-time recovery method in the recovery wizard when appropriate. The flash recovery area must be set to enable flashback database.

Flashback Retention Time: Hours

Current size of the flashback logs(GB): n/a

Lowest SCN in the flashback data: n/a

Flashback Time: n/a

Apply initialization parameter changes to SPFILE only. If not checked, parameter changes will be made to both the SPFILE and the running instance.

* Changes to this setting or parameter require a database restart.

Flash Recovery Area Usage

File Type	Size (GB)	Percentage
Online Log	0.15	2.5%
Backup Piece	0.08	1.3%
Archived Redo Log	0.05	0.8%
Control File	0.01	0.2%
Image Copy	0	0%
Flashback Log	0	0%
Usable	5.71	95.2%

Copyright © 2009, Oracle. Tous droits réservés.

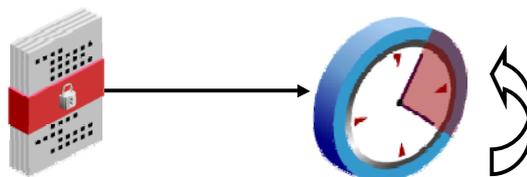
ORACLE

Surveiller Flashback Database avec EM

La plupart des statistiques Flashback Database mentionnées dans les pages précédentes peuvent être affichées à partir de la page Recovery Settings. Ces mesures de performances incluent l'espace actuel utilisé par tous les journaux Flashback, le numéro SCN le plus faible et l'heure correspondant au numéro SCN le plus faible dans les données flashback.

Points de restauration garantis

Les points de restauration garantis vous permettent d'exécuter la commande `FLASHBACK DATABASE` jusqu'au numéro SCN souhaité, à tout moment.



```
SQL> CREATE RESTORE POINT before_upgrade  
2 GUARANTEE FLASHBACK DATABASE;
```

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Points de restauration garantis

Tout comme les points de restauration normaux, les points de restauration garantis peuvent être utilisés comme des alias de numéros SCN dans les opérations de récupération. La principale différence réside dans le fait que les points de restauration garantis sont conservés sans limite de date dans le fichier de contrôle et doivent être supprimés de façon explicite. Ils offrent en outre une fonctionnalité spécifique liée à l'utilisation de la fonction Flashback Database.

Lorsque vous créez un point de restauration garanti pour un numéro SCN spécifique, vous êtes certain de pouvoir effectuer une opération Flashback Database visant à ramener la base de données à l'état correspondant à ce numéro SCN, même si la journalisation flashback n'est pas activée. Si la journalisation flashback est activée, la création d'un point de restauration garanti permet de mettre en oeuvre la conservation des journaux Flashback requis pour l'opération Flashback Database jusqu'à n'importe quel point dans le temps après la création du point de restauration garanti le plus ancien.

Vous pouvez utiliser un point de restauration garanti pour ramener la base de données entière à un état antérieur correct remontant à plusieurs jours ou à plusieurs semaines dès lors que l'espace disque disponible dans la zone de récupération rapide est suffisant pour le stockage des journaux requis. Tout comme les points de restauration normaux, les points de restauration garantis peuvent être utilisés pour indiquer un point dans le temps pour des opérations `RECOVER DATABASE`.

Remarque : Les limitations qui s'appliquent à Flashback Database s'appliquent également aux points de restauration garantis. Par exemple, la récupération d'espace dans un fichier de données ou la suppression d'un tablespace peut empêcher le flashback des fichiers de données affectés jusqu'au point de restauration garanti.

Flashback Database et points de restauration garantis

L'utilisation de points de restauration garantis nécessite que la base de données respecte les prérequis suivants :

- La valeur du paramètre `COMPATIBLE` doit être supérieure ou égale à 10.2.
- La base de données doit être exécutée en mode `ARCHIVELOG`.
- `FLASHBACK DATABASE` requiert l'utilisation de fichiers de journalisation archivés à compter de l'heure du point de restauration.
- Une zone de récupération rapide doit être configurée.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Flashback Database et points de restauration garantis

Pour prendre en charge l'utilisation de points de restauration garantis, la base de données doit satisfaire aux prérequis suivants :

- La valeur du paramètre `COMPATIBLE` doit être supérieure ou égale à 10.2.
- La base de données doit être exécutée en mode `ARCHIVELOG`.
- Pour le rembobinage de la base de données jusqu'à un point de restauration garanti, la commande `FLASHBACK DATABASE` requiert l'utilisation de fichiers de journalisation archivés (archived redo logs) à compter de l'heure du point de restauration.
- Une zone de récupération rapide doit être configurée. Les points de restauration garantis utilisent, en effet, un mécanisme similaire à la journalisation flashback. Comme pour cette dernière, la base de données Oracle doit stocker les journaux requis dans la zone de récupération rapide.
- Si la fonctionnalité Flashback Database n'est pas activée, la base de données doit être montée, mais non ouverte, lors de la création du premier point de restauration garanti (ou si tous ceux créés précédemment ont été supprimés).

Flashback Database et points de restauration garantis (suite)

La journalisation pour la fonctionnalité Flashback Database et les points de restauration garantis implique la capture d'images des blocs de fichiers de données avant application des modifications. La commande `FLASHBACK DATABASE` peut utiliser ces images pour rétablir l'état antérieur des fichiers de données. Les principales différences entre journalisation flashback classique, d'une part, et journalisation pour les points de restauration garantis, d'autre part, portent sur le moment où les blocs sont consignés et sur le fait que les journaux peuvent ou non être supprimés en réponse à un manque d'espace dans la zone de récupération rapide. Ces différences affectent l'utilisation de l'espace pour les journaux, ainsi que les performances de la base de données.

Si vous activez la fonctionnalité Flashback Database et que vous définissez un ou plusieurs points de restauration garantis, la base de données procède à une journalisation flashback classique. Dans ce cas, la zone de récupération conserve les journaux Flashback requis pour un flashback vers n'importe quel point arbitraire entre le présent et le point de restauration garanti le plus ancien actuellement défini. Les journaux Flashback ne sont pas supprimés en cas de manque d'espace s'ils sont nécessaires à la garantie.

Quiz

Vous pouvez configurer Flashback Database pour :

1. Réparer des corruptions logiques de données
2. Récupérer un tablespace qui a été supprimé
3. Effectuer une récupération jusqu'au moment où la taille d'un fichier de données a été réduite.
4. Effectuer une récupération jusqu'au moment où vous avez recréé le fichier de contrôle

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Réponse : 1

Quiz

Les journaux Flashback sont archivés pour permettre le rembobinage jusqu'à un point dans le temps qui n'est pas accessible à partir de la zone de récupération rapide.

1. Vrai
2. Faux

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Réponse : 2

Synthèse

Ce chapitre vous a permis d'apprendre à :

- configurer Flashback Database
- exécuter des opérations Flashback Database
- surveiller Flashback Database

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Présentation de l'exercice 12 : Utiliser Flashback Database

Cet exercice porte sur les points suivants :

- Effectuer une opération Flashback Database pour annuler des transactions non souhaitées
- Surveiller le délai de conservation pour Flashback Database
- Déterminer la taille des journaux Flashback

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

13

Gérer la mémoire

Copyright © 2009, Oracle. Tous droits réservés.

ORACLE

Laurent PERPETUE (laurent.perpetue@sir.fr) has a non-transferable license to use this Student Guide

Objectifs

A la fin de ce chapitre, vous pourrez :

- décrire les composants de la mémoire SGA
- implémenter la gestion automatique de la mémoire (ASMM)
- configurer manuellement les paramètres de la mémoire SGA
- configurer la gestion automatique de la mémoire PGA

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Gestion de la mémoire : Présentation

La gestion de la mémoire constitue l'un des points essentiels du travail d'un DBA, pour les raisons suivantes :

- La quantité de mémoire disponible est limitée.
- L'allocation de davantage de mémoire à certains types de fonction peut améliorer les performances globales.
- La configuration mise en oeuvre par le réglage automatique de l'allocation de mémoire convient généralement, mais certains environnements spécifiques ou des situations temporaires peuvent nécessiter une attention particulière.

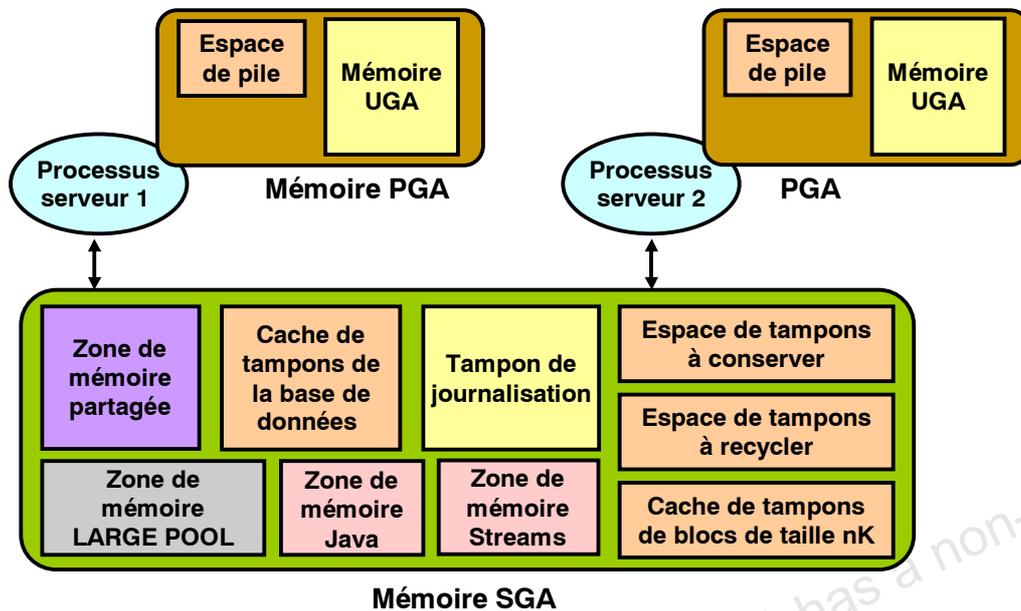
ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Gestion de la mémoire : Présentation

Parce que la quantité de mémoire disponible est limitée sur un serveur de base de données et, par conséquent, sur une instance de base de données Oracle, vous devez prêter une attention particulière à sa répartition. Si une quantité trop importante est allouée à une zone spécifique qui n'en a pas besoin, il est possible que d'autres zones fonctionnelles n'aient pas assez de mémoire pour fonctionner de manière optimale. L'allocation de mémoire pouvant être déterminée et gérée automatiquement, la tâche est grandement simplifiée. Toutefois, même en cas de réglage (tuning) automatique, il convient de surveiller la répartition de la mémoire afin de l'optimiser et il peut parfois être nécessaire d'effectuer une configuration manuelle partielle.

Rappels sur les structures mémoire d'une base Oracle



Copyright © 2009, Oracle. Tous droits réservés.

ORACLE

Rappels sur les structures mémoire d'une base Oracle

Oracle Database crée et utilise des structures mémoire à des fins diverses. La mémoire contient, par exemple, le code des programmes en cours d'exécution, les données partagées entre utilisateurs et des zones de données privées pour chaque utilisateur connecté.

A une instance sont associées deux structures mémoire élémentaires :

- **Mémoire SGA (System Global Area) :** Groupe de structures mémoire partagées (appelées composants SGA) qui contiennent les données et les informations de contrôle correspondant à une instance Oracle Database. Elle est partagée par l'ensemble des processus serveur et processus en arrière-plan. La mémoire SGA stocke des données telles que des blocs de données en cache et des zones SQL partagées.
- **Mémoire PGA (Program Global Area) :** Région de la mémoire qui contient des données et des informations de contrôle pour un processus serveur ou un processus en arrière-plan. Il s'agit d'une mémoire non partagée qui est créée par Oracle Database au démarrage d'un processus serveur ou processus en arrière-plan. Son accès est exclusivement réservé au processus serveur. Chaque processus serveur et processus en arrière-plan dispose de sa propre mémoire PGA.

Rappels sur les structures mémoire d'une base Oracle (suite)

La mémoire SGA est la zone de mémoire contenant les données et informations de contrôle relatives à l'instance. La mémoire SGA contient les structures de données suivantes :

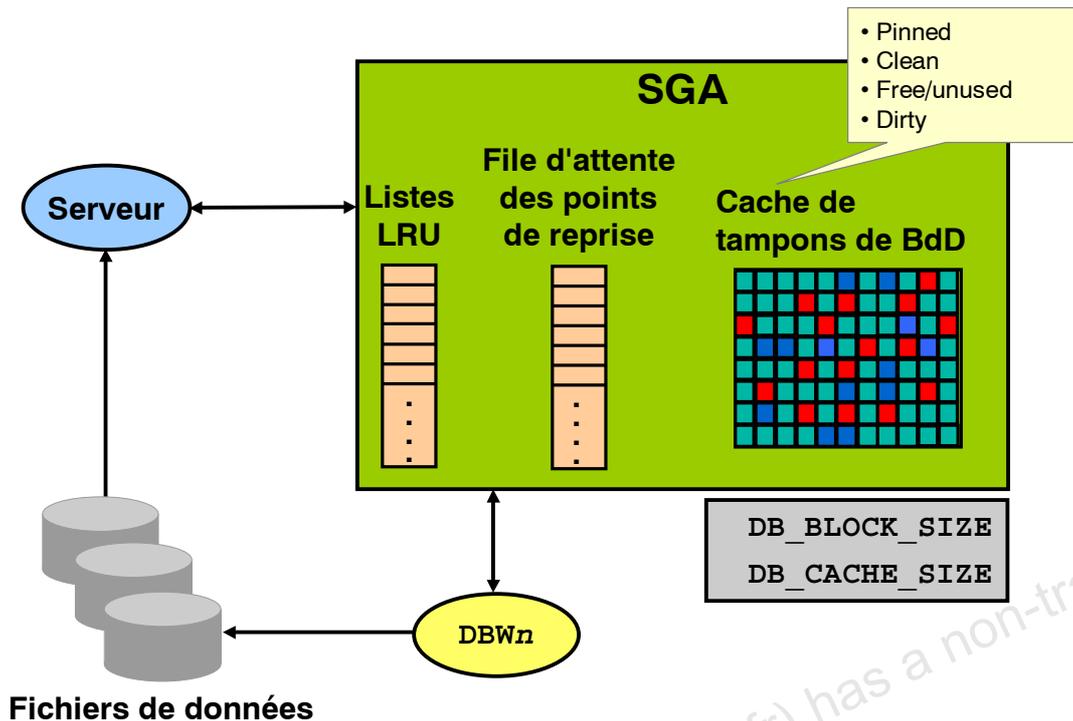
- **Zone de mémoire partagée** : Elle met en mémoire cache diverses structures pouvant être partagées par les utilisateurs.
- **Cache de tampons (buffer cache) de la base de données** : Il met en mémoire cache les blocs de données extraits de la base.
- **Espace de tampons à conserver** : Type particulier de cache de tampons qui est configuré pour le stockage de blocs de données pendant de longues périodes.
- **Espace de tampons à recycler** : Type particulier de cache de tampons qui est configuré pour un recyclage ou une suppression rapide de blocs de la mémoire.
- **Cache de tampons de blocs de taille nK** : Type particulier de cache de tampons conçu pour le stockage des blocs qui présentent une taille différente de la taille de bloc par défaut de la base de données.
- **Tampon de journalisation (redo log buffer)** : Il met en mémoire cache les informations de journalisation (utilisées pour la récupération de l'instance) jusqu'à ce qu'elles puissent être écrites dans les fichiers de journalisation physiques stockés sur le disque.
- **Zone de mémoire LARGE POOL** : Il s'agit d'une zone facultative fournissant d'importantes allocations de mémoire pour certains processus très consommateurs, tels que les opérations de sauvegarde et de récupération Oracle et les processus serveur d'entrée-sortie.
- **Zone de mémoire Java** : Elle est utilisée pour l'ensemble du code Java et des données propres à la session dans la JVM (Java Virtual Machine).
- **Zone de mémoire Streams** : Elle est utilisée par Oracle Streams pour stocker les informations nécessaires aux opérations de capture et d'application.

Lorsque vous démarrez l'instance via Enterprise Manager ou SQL*Plus, la quantité de mémoire allouée à la SGA est affichée.

Une mémoire PGA (Program Global Area) est une zone de mémoire contenant des données et des informations de contrôle pour chaque processus serveur. Un processus serveur Oracle traite les demandes d'un client. Chaque processus serveur dispose de sa propre zone de mémoire PGA privée, qui est créée lors du démarrage du processus serveur. L'accès à la mémoire PGA est exclusivement réservé à ce processus serveur, qui peut lire le contenu de la mémoire et écrire dans celle-ci par l'intermédiaire du code Oracle. La mémoire PGA est divisée en deux zones principales : l'espace de pile et la mémoire UGA (User Global Area). Grâce à l'infrastructure dynamique de la mémoire SGA, il est possible de modifier la taille du cache de tampons de la base de données, de la zone de mémoire partagée, de la zone de mémoire LARGE POOL, de la zone de mémoire Java et de la zone de mémoire Streams sans arrêter l'instance.

La base de données Oracle utilise des paramètres d'initialisation pour créer et gérer les structures mémoire. Le moyen le plus simple de gérer la mémoire est de permettre à la base de données de la gérer et de la régler automatiquement. Pour cela, il suffit (sur la plupart des plates-formes) de définir les paramètres d'initialisation MEMORY_TARGET et MEMORY_MAX_TARGET.

Cache de tampons



Copyright © 2009, Oracle. Tous droits réservés.

Cache de tampons

Vous pouvez configurer le cache de tampons en indiquant une valeur pour le paramètre `DB_CACHE_SIZE`. Les fichiers de données sont composés de blocs de données de taille `DB_BLOCK_SIZE`. Le cache de tampons contient des copies de ces blocs. Les blocs de données peuvent être partagés par tous les utilisateurs, car le cache de tampons fait partie de la mémoire SGA. Les processus serveur lisent le contenu des fichiers de données dans le cache de tampons. Pour de meilleures performances, ils lisent parfois plusieurs blocs en une seule opération. Les données contenues dans le cache de tampons sont écrites dans les fichiers de données par le processus `DBWn`. Pour de meilleures performances, ce processus copie plusieurs blocs en une seule opération.

A tout moment, le cache de tampons peut comporter plusieurs copies d'un même bloc de base de données. Seule une copie du bloc est effective, mais il arrive que les processus serveur doivent créer des copies cohérentes en lecture, à partir d'images du passé. On parle alors de bloc à lecture cohérente (consistent read (CR) block).

La liste LRU (Least Recently Used) reflète l'utilisation des mémoires tampons (buffers). Celles-ci sont triées en fonction de la fréquence et de l'ancienneté des références correspondantes. Par conséquent, les mémoires tampons utilisées le plus fréquemment et le plus récemment se trouvent à l'extrémité MRU (Most Recently Used). Les blocs entrants sont copiés dans une mémoire tampon à partir de l'extrémité LRU. Au départ, ce tampon est placé au milieu de la liste. Il est ensuite déplacé vers le haut ou vers le bas de la liste, selon sa fréquence d'utilisation.

Cache de tampons (suite)

Les mémoires tampons du cache de tampons peuvent présenter l'un des quatre états suivants :

- **Tampon faisant l'objet d'opérations pin (Pinned)** : Le bloc fait l'objet d'une opération de lecture ou d'écriture dans le cache. Les autres sessions attendent pour accéder au bloc.
- **Propre (Clean)** : La mémoire tampon ne fait plus l'objet d'opérations pin et elle est susceptible d'être immédiatement retirée si son contenu (bloc de données) n'est pas de nouveau référencé. Soit le contenu est synchronisé avec le disque, soit la mémoire tampon contient un cliché (snapshot) cohérent en lecture d'un bloc.
- **Disponible/inutilisé (Free/unused)** : La mémoire tampon est vide parce que l'instance vient juste de démarrer. Cet état est très similaire à l'état "clean", à ceci près que la mémoire tampon n'a pas été utilisée.
- **"Dirty"** : La mémoire tampon ne fait plus l'objet d'opérations pin mais le contenu (blocs de données) a été modifié et doit être transféré sur le disque par un processus DBWn avant son retrait de la mémoire.

Les processus serveur utilisent les mémoires tampons du cache de tampons. Pour faire en sorte que des mémoires tampons soient disponibles, le processus DBWn copie, dans les fichiers de données, le contenu de celles qui ont été modifiées. La file d'attente des points de reprise (checkpoints) répertorie les mémoires tampons devant faire l'objet d'une écriture sur le disque.

Une base de données Oracle prend en charge plusieurs tailles de bloc. La taille de bloc standard est utilisée pour le tablespace SYSTEM. Vous pouvez définir la taille de bloc standard via le paramètre d'initialisation DB_BLOCK_SIZE. Les valeurs autorisées sont comprises entre 2 ko et 32 ko. La valeur par défaut est 8 ko. La taille du cache des mémoires tampons présentant une taille de bloc non standard est définie par les paramètres suivants :

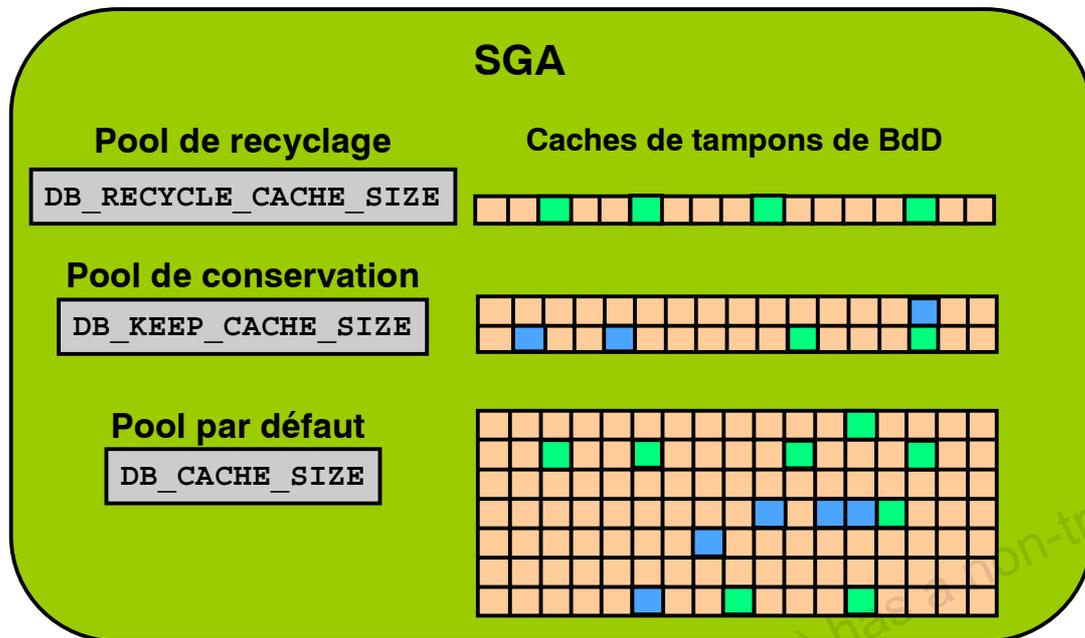
- DB_2K_CACHE_SIZE
- DB_4K_CACHE_SIZE
- DB_8K_CACHE_SIZE
- DB_16K_CACHE_SIZE
- DB_32K_CACHE_SIZE

Les paramètres DB_nK_CACHE_SIZE ne permettent pas de dimensionner le cache pour la taille de bloc standard. Si la valeur du paramètre DB_BLOCK_SIZE est nK, il n'est pas possible de définir DB_nK_CACHE_SIZE. La taille du cache pour la taille de bloc standard est toujours déterminée à partir de la valeur de DB_CACHE_SIZE.

Chaque cache de tampons présente une taille limitée. En général, les données du disque ne peuvent pas toutes tenir dans le cache. Lorsque le cache est plein et entraîne un échec, la base Oracle écrit sur le disque les données "dirty" déjà présentes dans celui-ci afin de libérer de l'espace pour les nouvelles données. Si une mémoire tampon n'est pas "dirty", il n'est pas nécessaire de l'écrire sur le disque pour permettre la lecture d'un nouveau bloc dans la mémoire tampon. Tout accès ultérieur aux données écrites sur le disque entraîne de nouveaux échecs dans le cache.

La taille du cache affecte la probabilité qu'une demande de données entraîne un succès en mémoire cache. Si la taille du cache est importante, la probabilité qu'il contienne les données demandées est plus élevée. Le fait d'augmenter la taille d'un cache augmente le pourcentage de demandes de données entraînant un succès en mémoire cache.

Utiliser des pools de tampons multiples



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Utiliser des pools de tampons multiples

L'administrateur de base de données (DBA) peut améliorer les performances du cache de tampons (buffer cache) de la base en créant des pools de tampons multiples. Les objets sont affectés à un pool de tampons en fonction de leur mode d'accès. Il existe trois pools de tampons :

- **Pool de conservation** : Ce pool est utilisé pour conserver en mémoire les objets qui seront probablement réutilisés. La conservation de ces objets en mémoire permet de réduire le nombre d'opérations d'E/S. Pour cela, il faut que le pool ait une taille supérieure à la taille totale des segments qui lui sont affectés. Ainsi, il n'est pas nécessaire que des mémoires tampons soient retirées de la mémoire (sur la base de la liste LRU). Le pool de conservation est configuré via l'affectation d'une valeur au paramètre `DB_KEEP_CACHE_SIZE`.
- **Pool de recyclage** : Ce pool est utilisé pour les blocs de mémoire dont la réutilisation est peu probable. Il est plus petit que la taille totale des segments qui lui sont affectés. Cela signifie que l'ajout d'un bloc dans le pool entraîne souvent le retrait d'un tampon de la mémoire. Le pool de recyclage est configuré via l'affectation d'une valeur au paramètre `DB_RECYCLE_CACHE_SIZE`.
- **Pool par défaut** : Ce pool existe toujours. Il équivaut au cache de tampons d'une instance sans pool de conservation ni pool de recyclage et est configuré via le paramètre `DB_CACHE_SIZE`.

Remarque : La mémoire du pool de conservation n'est pas un sous-ensemble du pool par défaut, pas plus que celle du pool de recyclage.

Utiliser des pools de tampons multiples

```
CREATE INDEX cust_idx ...  
  STORAGE (BUFFER_POOL KEEP);  
  
ALTER TABLE oe.customers  
  STORAGE (BUFFER_POOL RECYCLE);  
  
ALTER INDEX oe.cust_lname_ix  
  STORAGE (BUFFER_POOL KEEP);
```

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Utiliser des pools de tampons multiples (suite)

La clause `BUFFER_POOL` permet de définir le pool de tampons par défaut pour un objet. Elle fait partie de la clause `STORAGE` et peut être utilisée dans les instructions `CREATE` et `ALTER TABLE`, `CLUSTER` et `INDEX`. Les blocs provenant d'un objet pour lesquels aucun pool de tampons n'a été explicitement défini sont affectés au pool de tampons par défaut.

La syntaxe est la suivante : `BUFFER_POOL [KEEP | RECYCLE | DEFAULT]`.

Lorsque le pool de tampons par défaut d'un objet est modifié à l'aide de l'instruction `ALTER`, les blocs qui se trouvent déjà dans le cache restent dans leur mémoire tampon actuelle jusqu'à ce qu'ils soient vidés par l'activité de gestion normale du cache. Les blocs lus à partir du disque sont placés dans le nouveau pool de tampons désigné pour le segment.

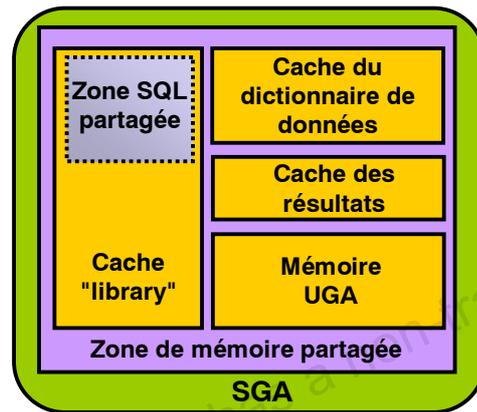
Les pools de tampons étant affectés à un segment, les objets comportant plusieurs segments peuvent avoir des blocs dans plusieurs pools. De même, une table organisée en index peut avoir des pools différents définis à la fois sur l'index et sur le segment de débordement (overflow segment).

Zone de mémoire partagée

Contenus :

- Cache "library" : instructions, code analysé et plans d'exécution.
- Cache du dictionnaire de données : définition des tables du dictionnaire de données, de leurs colonnes et des privilèges associés.
- Cache de résultats : résultats des interrogations SQL et des fonctions PL/SQL
- Mémoire UGA : informations de session pour Oracle Shared Server.

SHARED_POOL_SIZE



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Zone de mémoire partagée

Vous pouvez définir la taille de la zone de mémoire partagée à l'aide du paramètre d'initialisation `SHARED_POOL_SIZE`. Cette zone sert à stocker des informations partagées par plusieurs sessions. Elle contient différents types de données, comme le montre le schéma de la diapositive ci-dessus.

Cache "library" : Il contient des zones SQL et PL/SQL partagées, représentations entièrement analysées ou compilées de blocs PL/SQL et d'instructions SQL. Les blocs PL/SQL comprennent les éléments suivants :

- Procédures et fonctions
- Packages
- Déclencheurs (triggers)
- Blocs PL/SQL anonymes

Cache du dictionnaire de données : Il contient les définitions des objets du dictionnaire en mémoire.

Cache de résultats : Il contient les résultats des interrogations SQL et des fonctions PL/SQL. Le traitement est ainsi accéléré lorsqu'elles sont réexécutées.

Mémoire UGA : Elle contient les informations de session relatives à Oracle Shared Server. Elle se trouve dans la zone de mémoire partagée lors de l'utilisation d'une session de serveur partagé et lorsque la zone de mémoire `LARGE POOL` n'est pas configurée.

Zone de mémoire LARGE POOL

- Elle fournit des allocations de mémoire de grande taille pour les éléments suivants :
 - Mémoire de session pour le serveur partagé et l'interface Oracle XA
 - Processus serveur d'E/S
 - Opérations de sauvegarde et de restauration d'Oracle Database
 - Opérations Parallel Query
 - Mémoire Advanced Queuing
- Elle permet de limiter la fragmentation de la zone de mémoire partagée.
- Elle est gérée par AMM et ASMM.
- Sa taille est définie par le paramètre `LARGE_POOL_SIZE`.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Zone de mémoire LARGE POOL

L'administrateur de base de données peut configurer une zone de mémoire facultative appelée *LARGE POOL* pour allouer des zones de mémoire de grande taille aux éléments suivants :

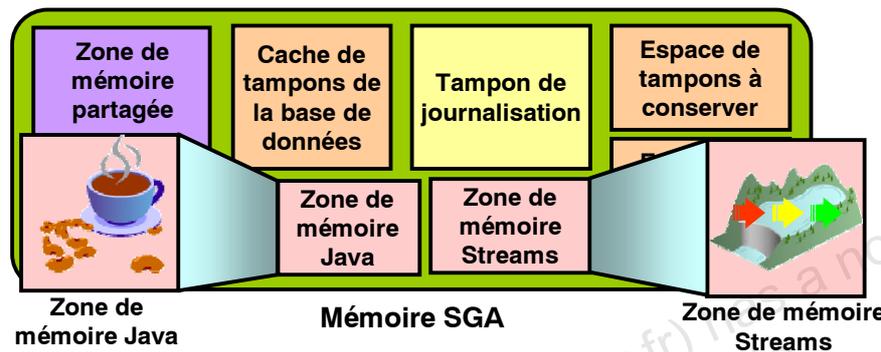
- Mémoire de session pour le serveur partagé et l'interface Oracle XA (lorsque les transactions interagissent avec plusieurs bases de données)
- Processus serveur d'E/S
- Mémoires tampons pour les esclaves d'E/S Recovery Manager (RMAN)
- Tampons de messages utilisés pour l'exécution parallèle d'instructions
- Mémoire Advanced Queuing

Grâce à l'allocation de mémoire de session aux éléments répertoriés dans la diapositive, la zone de mémoire partagée subit moins de fragmentation que si les objets volumineux y étaient fréquemment alloués et désalloués. Le fait de placer les objets volumineux en dehors de la zone de mémoire partagée permet d'optimiser l'utilisation de celle-ci. Une part plus importante de cette mémoire est disponible pour le traitement de nouvelles demandes et pour la conservation des données existantes en cas de besoin.

La zone de mémoire *LARGE POOL* peut être gérée automatiquement avec AMM et ASMM. Sa taille est définie par le paramètre `LARGE_POOL_SIZE`.

Zones de mémoire Java et Streams

- La zone de mémoire Java est utilisée pour l'ensemble du code Java et des données propres à la session dans la JVM (Java Virtual Machine).
- La zone de mémoire Streams est utilisée exclusivement par Oracle Streams :
 - Stockage des messages de la file d'attente tampon
 - Fourniture de mémoire pour les processus Oracle Streams



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Zones de mémoire Java et Streams

La zone de mémoire Java est utilisée pour l'ensemble du code Java et des données propres à la session dans la JVM (Java Virtual Machine). Elle est utilisée de différentes manières, en fonction du mode d'exécution d'Oracle Database.

Les statistiques générées par Java Pool Advisor fournissent des informations sur le cache "library" utilisé pour Java et indiquent comment les changements de taille de la zone de mémoire Java peuvent affecter le taux d'analyse. Java Pool Advisor est activé en interne lorsque `statistics_level` a la valeur `TYPICAL` ou une valeur supérieure. Lorsqu'il est désactivé, les statistiques sont réinitialisées.

La zone de mémoire Streams est utilisée par Oracle Streams exclusivement. Elle contient les messages en file d'attente tampon et fournit la mémoire nécessaire aux processus Oracle Streams de capture et d'application des modifications.

La taille de la zone de mémoire Streams est initialement égale à zéro (sauf configuration spécifique). Elle augmente dynamiquement en fonction des besoins quand Oracle Streams est utilisé.

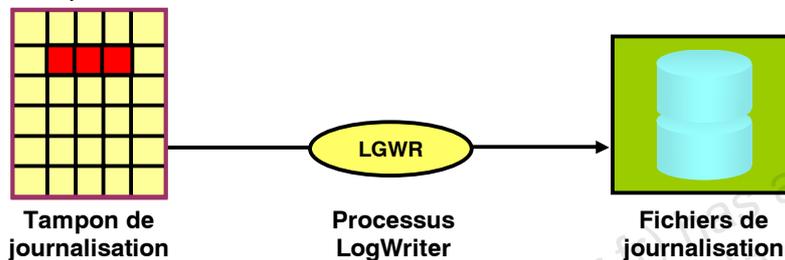
Remarque : Ce cours ne traite pas en détail de la programmation Java ni d'Oracle Streams.

Tampon de journalisation

- Mémoire tampon réutilisable située dans la mémoire SGA
- Contient des informations sur les modifications apportées à la base de données
- Contient des entrées de journalisation permettant d'annuler les modifications effectuées par des opérations LMD et LDD

Contenu transféré par le processus LGWR (Log Writer) :

- lorsqu'un processus utilisateur valide une transaction
- quand un tiers du tampon de journalisation est plein
- avant qu'un processus DBWn n'écrive les tampons modifiés sur le disque



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Tampon de journalisation

Les processus serveur Oracle copient les entrées de journalisation de l'espace mémoire utilisateur dans le tampon de journalisation, et ce pour chaque instruction LMD ou LDD. Les entrées de journalisation contiennent les informations nécessaires pour reconstruire ou réappliquer les modifications apportées à la base de données par les opérations LMD et LDD. Elles sont utilisées pour la récupération de la base et occupent un espace continu et séquentiel dans la mémoire tampon.

Le tampon de journalisation est circulaire. Les processus serveur peuvent copier de nouvelles entrées en écrasant celles qui ont déjà été écrites sur le disque. En principe, le processus LGWR écrit suffisamment rapidement pour garantir aux nouvelles entrées de l'espace disponible dans la mémoire tampon. Il écrit le tampon de journalisation dans le fichier de journalisation en ligne actif (ou les membres du groupe actif) sur le disque. Il copie sur le disque toutes les entrées de journalisation qui ont été introduites dans la mémoire tampon depuis la dernière opération d'écriture sur le disque par le processus LGWR.

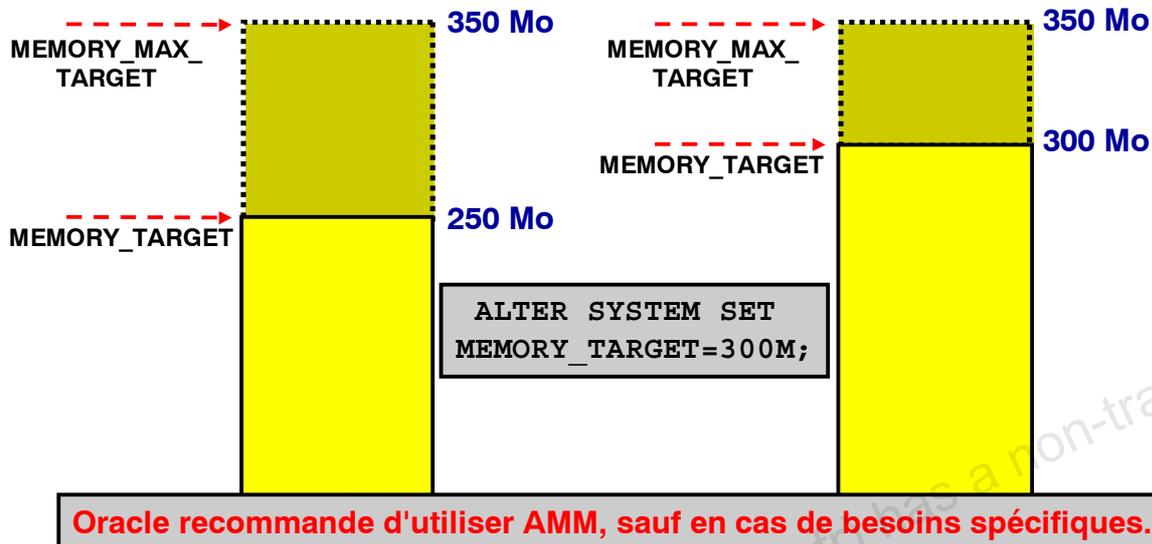
Événements qui déclenchent une opération d'écriture par le processus LGWR

Le processus LGWR efface les données du tampon de journalisation :

- lorsqu'un processus utilisateur valide (commit) une transaction,
- toutes les trois secondes ou quand un tiers du tampon de journalisation est plein,
- lorsqu'un processus DBWn écrit des mémoires tampons modifiées sur le disque, si les données de journalisation correspondantes n'ont pas encore été écrites sur le disque.

Gestion automatique de la mémoire : Présentation

Avec la gestion automatique de la mémoire, la base de données peut dimensionner les mémoires SGA et PGA automatiquement en fonction de la charge globale.



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Gestion automatique de la mémoire : Présentation

La gestion automatique de la mémoire (AMM) permet à la base de données Oracle de gérer automatiquement le dimensionnement de la mémoire SGA et de la mémoire PGA de l'instance. Pour ce faire (sur la plupart des plates-formes), vous définissez uniquement un paramètre d'initialisation pour la taille de mémoire cible (MEMORY_TARGET) et un paramètre d'initialisation pour la taille de mémoire maximum (MEMORY_MAX_TARGET).

La base de données échange ensuite de la mémoire de façon dynamique entre la mémoire SGA et la mémoire PGA de l'instance afin de répondre aux demandes de traitement.

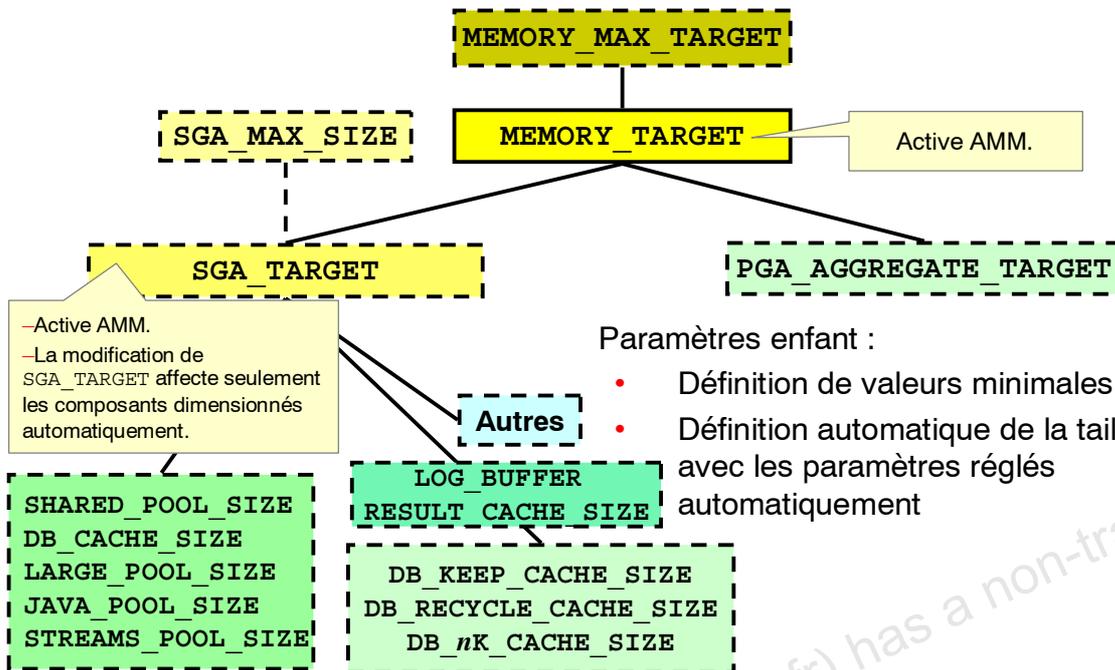
Vous pouvez activer AMM dans Enterprise Manager en sélectionnant Server > Memory Advisors (dans la section Database Configuration) puis en cliquant sur le bouton Enable.

Avec cette méthode de gestion de la mémoire, la base de données règle aussi automatiquement la taille des différents composants de la mémoire SGA et celle des différentes mémoires PGA.

Le paramètre d'initialisation de la mémoire cible étant dynamique, vous pouvez modifier à tout moment la taille de la mémoire cible sans redémarrer la base de données. Le paramètre de taille de mémoire maximum sert de limite supérieure afin que vous ne puissiez pas attribuer de valeur trop élevée à la taille de mémoire cible. Puisqu'il est difficile de réduire la taille de certains composants de la mémoire SGA ou que certains d'entre eux doivent présenter une taille minimum, la base de données vous empêche également d'indiquer une taille de mémoire cible trop faible.

Ce transfert indirect de mémoire repose sur le mécanisme de libération de mémoire partagée du système d'exploitation. Après restitution d'espace mémoire au système d'exploitation, les autres composants peuvent effectuer des demandes d'allocation de mémoire. La gestion automatique de la mémoire est actuellement implémentée sous Linux, Solaris, HP-UX, AIX et Windows.

Paramètres de dimensionnement de la mémoire Oracle Database



Copyright © 2009, Oracle. Tous droits réservés.

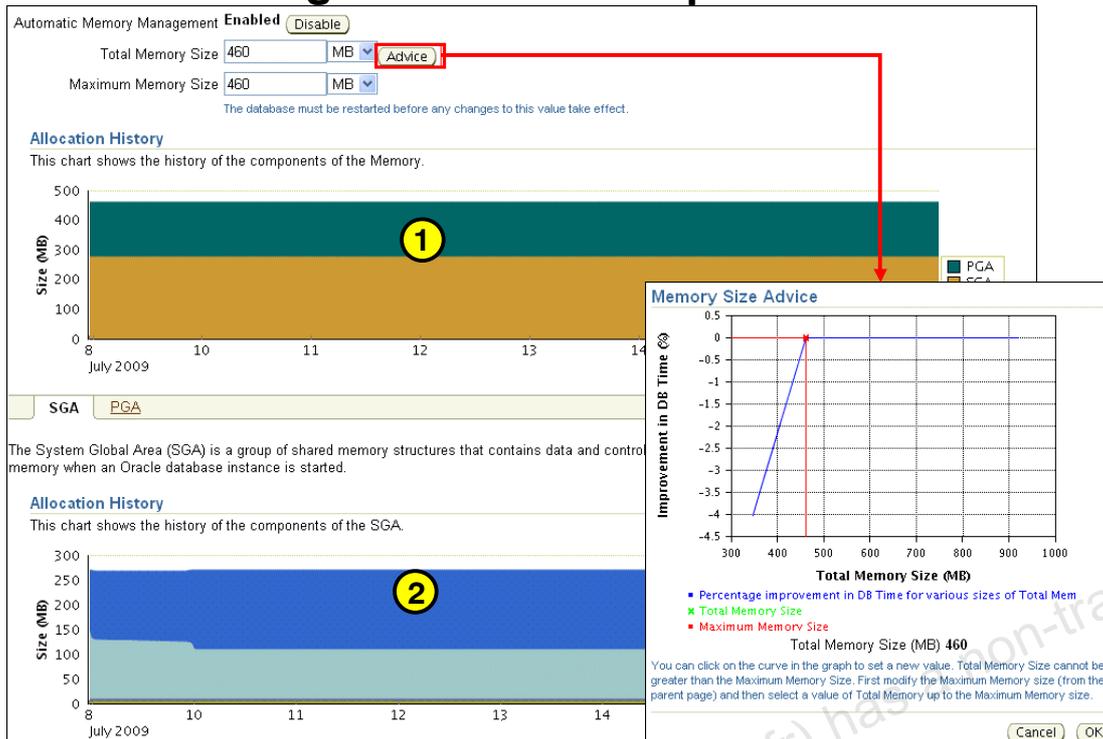
ORACLE

Paramètres de dimensionnement de la mémoire Oracle Database

Le schéma de la diapositive ci-dessus illustre la hiérarchie des paramètres d'initialisation relatifs à la mémoire. Il suffit de définir MEMORY_TARGET pour déclencher la gestion automatique de la mémoire. Toutefois, vous avez toujours la possibilité de définir des limites inférieures pour différents caches. Par conséquent, si des valeurs sont définies par l'utilisateur pour les paramètres enfant, il s'agit de valeurs minimum en dessous desquelles le serveur de base de données Oracle ne procède pas à un réglage automatique du composant correspondant.

- Si les paramètres SGA_TARGET et PGA_AGGREGATE_TARGET sont définis avec une valeur différente de zéro, ils sont considérés comme les valeurs minimales pour la taille des mémoires SGA et PGA (respectivement). MEMORY_TARGET peut additionner les valeurs des paramètres SGA_TARGET et PGA_AGGREGATE_TARGET pour obtenir MEMORY_MAX_SIZE.
- Si SGA_TARGET est défini, la base de données procède uniquement au réglage automatique de la taille des sous-composants de la mémoire SGA. La mémoire PGA fait l'objet d'un réglage automatique qu'elle soit définie ou non de façon explicite. Toutefois, les mémoires SGA (SGA_TARGET) et PGA (PGA_AGGREGATE_TARGET) ne font pas l'objet d'un réglage automatique, c'est-à-dire que leur taille n'augmente ni ne diminue automatiquement.

Surveiller la gestion automatique de la mémoire



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Surveiller la gestion automatique de la mémoire

Dans la page d'accueil EM (section Related Links), sélectionnez Advisor Central > Memory Advisors. La page Memory Advisors est affichée dans la diapositive.

Une fois la gestion automatique de la mémoire activée, vous pouvez consulter une représentation graphique de l'historique des composants de taille de mémoire dans la section Allocation History de la page Memory Advisors. La partie supérieure du premier histogramme correspond uniquement à la mémoire PGA réglable, et la partie inférieure, à l'ensemble de la mémoire SGA. La partie supérieure du deuxième histogramme correspond à la taille de la zone de mémoire partagée, et la partie inférieure, au cache de tampons (buffer cache).

Sur cette page, vous pouvez aussi accéder à la fonction de conseil Memory Target Advisor en cliquant sur le bouton Advice. Cette fonction vous indique l'amélioration possible du temps d'exécution de la base de données en fonction de différentes tailles de mémoire totale.

Remarque : Vous pouvez également consulter la fonction Memory Target Advisor à l'aide de la vue `V$MEMORY_TARGET_ADVISOR`.

Surveiller la gestion automatique de la mémoire

Vous pouvez aussi surveiller les actions entreprises par la fonction de gestion automatique de la mémoire à partir d'une ligne de commande :

- La vue `V$MEMORY_DYNAMIC_COMPONENTS` indique le statut en cours de tous les composants de mémoire.
- La vue `V$MEMORY_RESIZE_OPS` présente une mémoire tampon réutilisable des 800 dernières demandes de redimensionnement de mémoire.
- La vue `V$MEMORY_TARGET_ADVICE` fournit des conseils de réglage concernant le paramètre d'initialisation `MEMORY_TARGET`.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Surveiller la gestion automatique de la mémoire (suite)

La vue dynamique des performances `V$MEMORY_DYNAMIC_COMPONENTS` indique la taille actuelle de tous les composants de mémoire réglés de façon dynamique, y compris les tailles totales de la mémoire SGA et de la mémoire PGA de l'instance. La vue `V$MEMORY_TARGET_ADVICE` fournit des conseils de réglage concernant le paramètre d'initialisation `MEMORY_TARGET`.

Dans cette vue, la ligne sur laquelle `MEMORY_SIZE_FACTOR` est égal à 1 indique la taille de mémoire actuelle, comme défini par le paramètre d'initialisation `MEMORY_TARGET`, ainsi que le temps d'exécution de la base de données nécessaire à l'accomplissement de la charge globale actuelle. Dans les lignes précédentes et suivantes, les résultats indiquent un certain nombre de tailles `MEMORY_TARGET` alternatives. Pour chaque taille alternative, la base de données indique le facteur de taille (multiple de la taille actuelle), ainsi qu'une estimation du temps d'exécution nécessaire à l'accomplissement de la charge globale actuelle si le paramètre `MEMORY_TARGET` se voyait attribuer cette taille alternative. Notez que si la taille de mémoire totale est inférieure à la taille `MEMORY_TARGET` actuelle, l'estimation du temps d'exécution de la base de données diminue.

Utilisation efficace de la mémoire : Recommandations

- Essayez de faire tenir la mémoire SGA dans la mémoire physique.
- Effectuez un réglage de façon à obtenir un taux élevé de succès en mémoire cache de tampons, en tenant compte des points suivants :
 - Même les balayages complets de table valides et nécessaires font baisser le taux.
 - Il est possible que des lectures répétées inutiles des mêmes blocs fassent monter artificiellement le taux.
- Utilisez la fonction de conseil Memory Advisor.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Utilisation efficace de la mémoire : Recommandations

Si possible, il vaut mieux faire tenir la mémoire SGA dans la mémoire physique, car celle-ci permet des accès plus rapides. Même si le système d'exploitation peut fournir de la mémoire virtuelle supplémentaire, cette mémoire est susceptible de faire l'objet d'un swap sur le disque, de par sa nature. Sur certaines plates-formes, vous pouvez utiliser le paramètre d'initialisation `LOCK_SGA` pour verrouiller la mémoire SGA dans la mémoire physique. Ce paramètre ne peut pas être utilisé conjointement avec la fonction AMM ou ASMM.

Lorsqu'une instruction SQL s'exécute, les blocs de données font l'objet d'une demande de lecture ou d'écriture, ou les deux. On considère cela comme une E/S logique. Lorsqu'un bloc est sollicité, le système vérifie d'abord s'il existe déjà en mémoire. S'il n'est pas en mémoire, il est lu à partir du disque. On appelle cela une E/S physique. Le taux de succès en mémoire cache de tampons (buffer cache hit ratio) définit le nombre de fois où le bloc est déjà en mémoire comparé au nombre total d'E/S logiques. Un taux de succès élevé est généralement préférable car cela signifie que davantage de blocs sont détectés en mémoire sans générer d'E/S disque.

Il n'est pas rare d'avoir un taux de succès en mémoire cache de tampons supérieur à 99 %, mais cela ne signifie pas toujours que le système est bien réglé. Si une interrogation est exécutée plus souvent que nécessaire, sollicitant constamment les mêmes blocs de façon répétée, le taux augmente. Si l'interrogation en question est inefficace ou inutile, elle fait monter artificiellement le taux. Cela signifie qu'elle ne devrait pas être exécutée de cette façon ou aussi souvent.

Utilisation efficace de la mémoire : Recommandations (suite)

Les grands balayages complets de table (full table scans) (lecture complète d'une table entière) peuvent également faire baisser le taux de succès en mémoire cache de tampons, car ce type de lecture peut s'effectuer à partir du disque. Le processus de balayage peut ne pas tenir compte du fait que certains blocs se trouvent dans le cache de tampons (buffer cache). Par conséquent, si d'importants balayages complets de table sont nécessaires dans l'application que vous utilisez, le taux de succès en mémoire cache de tampons peut demeurer bas, même si vous avez effectué les réglages appropriés.

Utilisez la fonction de conseil Memory Advisor d'Enterprise Manager. Cette fonction vous permet de dimensionner la mémoire SGA en fonction de l'activité de la base.

Recommandations relatives au réglage de la mémoire pour le cache "library"

- Etablissez des conventions de formatage pour les développeurs afin que les instructions SQL correspondent dans le cache.
- Utilisez des variables attachées.
- Eliminez les instructions SQL dupliquées inutiles.
- Pensez à utiliser le paramètre `CURSOR_SHARING`.
- Utilisez le package PL/SQL lorsque cela est possible.
- Mettez en mémoire cache les numéros de séquence.
- Effectuez une opération pin sur les objets dans le cache "library".

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Recommandations relatives au réglage de la mémoire pour le cache "library"

Le cache "library", qui fait partie de la zone de mémoire partagée, est le lieu où la base de données Oracle stocke l'ensemble des instructions SQL, des codes Java, des procédures et packages PL/SQL, et des structures de contrôle telles que les verrous externes (locks) et les pointeur de cache "library". Une fois stocké dans cet emplacement central, le code peut être partagé par l'ensemble des utilisateurs. L'avantage de ce partage est que tous les utilisateurs peuvent tirer parti du travail déjà effectué par l'intermédiaire du code SQL. Par conséquent, les tâches consistant par exemple à analyser l'instruction et à déterminer le chemin d'accès aux données (outil Explain Plan) ne sont réalisées qu'une seule fois par instruction, quels que soient le nombre d'exécutions de l'instruction et le nombre d'utilisateurs qui l'exécutent. Un cache "library" trop petit ne peut pas contenir toutes les instructions exécutées. Dans un tel cas, les avantages du partage sont limités à certaines instructions. A l'inverse, un cache "library" trop grand impose une gestion lourde qui constitue une charge pour le système.

Il se peut que le cache "library" semble contenir des instructions distinctes, alors qu'il s'agit en réalité de copies d'instructions identiques. Cette situation résulte généralement d'un formatage légèrement différent pour les instructions. Dans ce cas, elles ne sont pas considérées comme identiques par l'opération de comparaison. Cette situation peut également résulter de l'utilisation de valeurs littérales au lieu de variables attachées (bind variables). Si l'utilisation de valeurs littérales est le seul élément qui différencie deux instructions, il convient généralement de remplacer ces valeurs par des variables attachées pour la bonne exécution des instructions et pour le bon fonctionnement du système en général.

Recommandations relatives au réglage de la mémoire pour le cache "library" (suite)

Le paramètre d'initialisation `CURSOR_SHARING` peut être défini de façon à ce que le système remplace automatiquement toutes les valeurs littérales par des variables attachées (bind variables) lorsque les instructions ne présentent pas d'autres différences. Il est recommandé d'utiliser ce paramétrage en tant que mesure temporaire jusqu'à ce que l'application soit corrigée de manière à utiliser des variables attachées de façon appropriée. D'une manière générale, l'utilisation de variables peut avoir d'autres effets, que vous devez identifier.

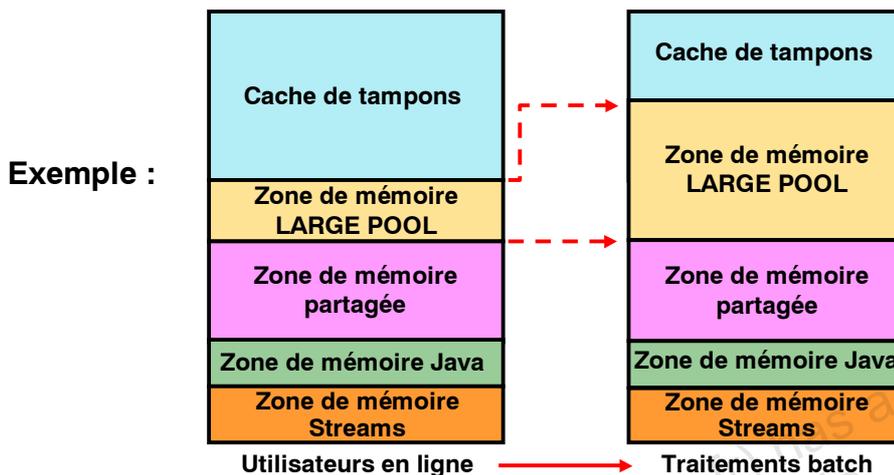
Au lieu d'exécuter la même instruction SQL à partir de plusieurs points différents d'une application, placez l'instruction ou les instructions souhaitées dans une procédure stockée en utilisant le langage PL/SQL. Appelez ensuite cette procédure. Vous garantissez ainsi que l'instruction SQL est partagée car elle n'existe qu'à un seul endroit. De plus, l'instruction SQL est déjà analysée et comporte un Explain Plan car elle se trouve dans une procédure stockée déjà compilée.

Les numéros de séquence peuvent être mis en mémoire cache. Par conséquent, si certaines séquences sont l'objet d'une activité importante, déterminez un paramétrage approprié pour la taille du cache afin d'en tirer parti.

Vous pouvez utiliser le package `DBMS_SHARED_POOL` pour effectuer une opération pin sur les objets dans le cache "library". Cette opération réduit la probabilité de rechargement et de recompilation des objets. Reportez-vous au document *PL/SQL Packages and Types Reference* pour plus d'informations sur l'utilisation de ce package.

Gestion automatique de la mémoire partagée : Présentation

- Permet une adaptation automatique aux variations de la charge globale
- Optimise l'utilisation de la mémoire
- Contribue à éviter les erreurs liées à une mémoire insuffisante



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Gestion automatique de la mémoire partagée : Présentation

Si vous ne pouvez pas utiliser AMM car vous avez besoin d'une mémoire PGA fixe, envisagez de recourir à ASMM (Automatic Shared Memory Management) qui simplifie la gestion de la mémoire SGA. Le paramètre d'initialisation `SGA_TARGET` vous permet d'indiquer la quantité totale de mémoire SGA à la disposition d'une instance et la base de données Oracle distribue automatiquement cette mémoire entre les différents composants de la mémoire SGA afin d'en garantir l'utilisation la plus efficace possible.

Par exemple, dans un système qui exécute des travaux OLTP (Traitement des transactions en ligne) volumineux pendant la journée (nécessitant ainsi un cache de tampons volumineux) et qui exécute des traitements batch en parallèle pendant la nuit (nécessitant une valeur importante pour la zone de mémoire LARGE POOL), vous devez configurer simultanément le cache de tampons et la zone de mémoire LARGE POOL afin de pouvoir faire face aux pics de charge.

Avec la fonction ASMM, lorsque le travail OLTP s'exécute, le cache de tampons utilise la majeure partie de la mémoire afin de garantir des performances d'E/S optimales. Lorsque le traitement batch d'analyse de données et de création d'états démarre par la suite, la mémoire est automatiquement affectée à la zone de mémoire LARGE POOL, afin de pouvoir être utilisée par les opérations d'interrogation en parallèle sans provoquer d'erreurs de débordement de mémoire.

Si vous utilisez un fichier de paramètres serveur (SPFILE), la base de données Oracle se souvient de la taille des composants qui font l'objet d'un réglage automatique, même après un arrêt de l'instance. Ainsi, le système n'a pas besoin de prendre à nouveau connaissance des caractéristiques de la charge globale à chaque démarrage d'une instance. Il peut utiliser comme point de départ les informations de l'instance précédente et poursuivre l'évaluation de la charge globale là où il s'était arrêté.

Fonctionnement de la gestion automatique de la mémoire partagée

- La gestion automatique de la mémoire partagée est basée sur des informations relatives à la charge globale que MMON capture en arrière-plan.
- MMON utilise des conseils sur la mémoire.
- La mémoire est déplacée là où le système en a le plus besoin.
- Si un fichier SPFILE est utilisé (ce qui est recommandé) :
 - La taille des composants est enregistrée lors des arrêts.
 - Les valeurs enregistrées sont utilisées pour initialiser (bootstrap) la taille des composants.
 - Il n'est pas nécessaire de redéterminer les valeurs optimales.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Fonctionnement de la gestion automatique de la mémoire partagée

La fonction ASMM (Automatic Shared Memory Management) utilise l'interface Broker de la mémoire SGA qui est implémentée par deux processus en arrière-plan : Manageability Monitor (MMON) et Memory Manager (MMAN). Les statistiques et les données de conseil sur la mémoire sont capturées périodiquement dans la mémoire par le processus MMON. MMAN coordonne le dimensionnement des composants de mémoire en fonction des décisions de MMON. Cette interface Broker assure le suivi de la taille des composants et des opérations de redimensionnement en attente.

L'interface Broker de la mémoire SGA observe le système et la charge globale afin de déterminer la distribution idéale de la mémoire. Elle effectue cette vérification chaque minute, afin que la mémoire soit toujours disponible lorsqu'elle est nécessaire.

Avant l'introduction de la fonctionnalité de gestion automatique de la mémoire partagée, les composants devaient être dimensionnés pour anticiper les besoins en mémoire dans les scénarios les plus défavorables.

A partir des informations sur la charge globale, la gestion automatique de la mémoire partagée :

- capture périodiquement des statistiques en arrière-plan,
- utilise des conseils sur la mémoire (memory advisors),
- effectue une analyse de simulation afin de déterminer la meilleure distribution de la mémoire,
- déplace la mémoire là où le système en a le plus besoin,
- enregistre la taille des composants lors des arrêts si un fichier SPFILE est utilisé (il est ainsi possible de récupérer les tailles utilisées avant le dernier arrêt).

Activer la gestion automatique de la mémoire partagée

Pour activer la fonction ASMM à partir d'une gestion manuelle de la mémoire partagée :

1. Obtenez une valeur pour `SGA_TARGET` :

```
SELECT ((SELECT SUM(value) FROM V$SGA) - (SELECT CURRENT_SIZE FROM V$SGA_DYNAMIC_FREE_MEMORY)) "SGA_TARGET" FROM DUAL;
```

2. Utilisez cette valeur pour définir `SGA_TARGET`.
3. Attribuez la valeur zéro aux composants de la mémoire SGA faisant l'objet d'un dimensionnement automatique.

Pour passer de la fonction AMM à la fonction ASMM :

1. Attribuez au paramètre d'initialisation `MEMORY_TARGET` la valeur 0.
2. Attribuez la valeur zéro aux composants de la mémoire SGA faisant l'objet d'un dimensionnement automatique.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Activer la gestion automatique de la mémoire partagée

La procédure d'activation de la fonction ASMM diffère selon que vous utilisiez auparavant une gestion manuelle de la mémoire partagée ou une gestion automatique de la mémoire.

Pour passer à la fonction ASMM à partir d'une gestion manuelle de la mémoire partagée :

1. Exécutez l'interrogation suivante pour obtenir une valeur pour `SGA_TARGET` :

```
SELECT ((SELECT SUM(value) FROM V$SGA) - (SELECT CURRENT_SIZE FROM V$SGA_DYNAMIC_FREE_MEMORY)) "SGA_TARGET" FROM DUAL;
```

2. Définissez la valeur de `SGA_TARGET` :

```
ALTER SYSTEM SET SGA_TARGET=value [SCOPE={SPFILE|MEMORY|BOTH}]
```

où *value* est la valeur calculée à l'étape 1 ou une valeur comprise entre la somme des tailles de tous les composants de la mémoire SGA et `SGA_MAX_SIZE`.

3. Affectez la valeur 0 aux composants de la SGA faisant l'objet d'un dimensionnement automatique. Pour cela, modifiez le fichier texte des paramètres d'initialisation ou exécutez les instructions `ALTER SYSTEM`. Redémarrez l'instance si nécessaire.

Pour passer à la fonction ASMM à partir d'une gestion automatique de la mémoire :

1. Attribuez au paramètre d'initialisation `MEMORY_TARGET` la valeur 0.

```
ALTER SYSTEM SET MEMORY_TARGET = 0;
```

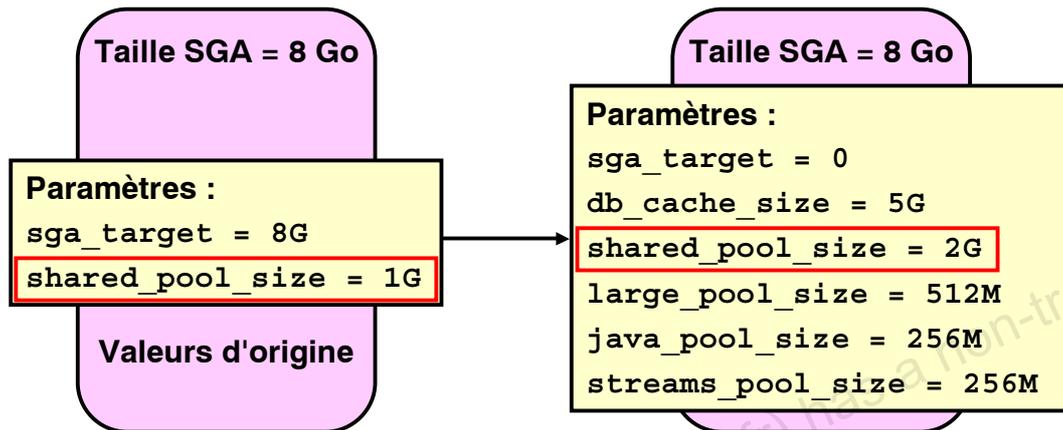
La base de données définit `SGA_TARGET` en fonction de l'allocation de mémoire SGA actuelle.

2. Attribuez la valeur zéro aux composants de la mémoire SGA faisant l'objet d'un dimensionnement automatique. Redémarrez ensuite l'instance.

Remarque : La fonction Automatic Memory Management sera décrite ultérieurement dans ce chapitre.

Désactiver la fonction ASMM

- L'affectation de la valeur zéro à `SGA_TARGET` désactive le réglage automatique.
- Les paramètres à réglage automatique sont définis avec leur taille actuelle.
- La taille globale de la mémoire SGA n'est pas affectée.



ORACLE

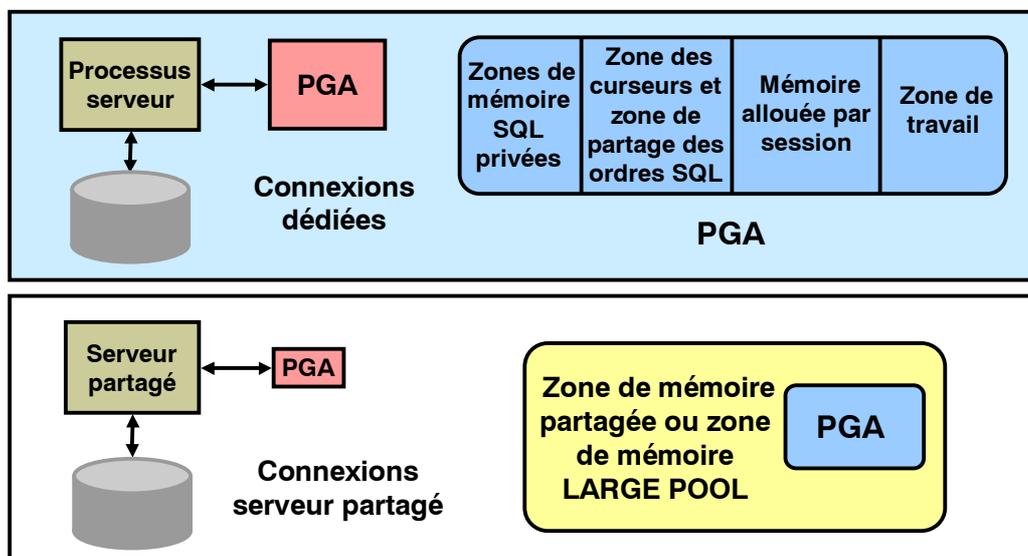
Copyright © 2009, Oracle. Tous droits réservés.

Désactiver la fonction ASMM

Vous pouvez désactiver la fonction ASMM en affectant la valeur 0 à `SGA_TARGET`. Dans ce cas, tous les paramètres à réglage automatique prennent une valeur égale à la taille actuelle des composants correspondants, même si l'utilisateur avait affecté antérieurement une autre valeur différente de zéro à l'un d'eux.

Dans l'exemple de la diapositive ci-dessus, la valeur du paramètre `SGA_TARGET` est de 8 Go et celle du paramètre `SHARED_POOL_SIZE` est de 1 Go. Si le système a ajusté en interne la taille de la zone de mémoire partagée à 2 Go, le fait d'affecter la valeur zéro au paramètre `SGA_TARGET` entraîne l'affectation d'une valeur de 2 Go au paramètre `SHARED_POOL_SIZE`, qui remplace la valeur définie à l'origine par l'utilisateur.

Mémoire PGA



Par défaut, la gestion automatique de la mémoire PGA est activée.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Mémoire PGA

La mémoire PGA (Program Global Area) est une zone de mémoire contenant des données et des informations de contrôle pour un processus serveur. Il s'agit d'une mémoire non partagée qui est créée par le serveur Oracle au démarrage d'un processus serveur. Son accès est limité à ce processus serveur. La mémoire PGA totale allouée par tous les processus serveur associés à une instance Oracle est également appelée *mémoire PGA agrégée* allouée par l'instance.

Une partie de la mémoire PGA peut résider dans la mémoire SGA lors de l'utilisation de serveurs partagés.

La mémoire PGA contient généralement les éléments suivants :

Zone de mémoire SQL privée

Une zone de mémoire SQL privée contient des données telles que des informations d'attachement et des structures mémoire d'exécution. Ces informations sont propres à l'appel de l'instruction SQL de chaque session. Ainsi, les variables attachées (bind variables) contiennent des valeurs différentes et l'état du curseur est différent. Chaque session qui exécute une instruction SQL dispose d'une zone de mémoire SQL privée. La zone de mémoire SQL privée de chaque utilisateur qui soumet une même instruction SQL utilise une même zone SQL partagée. Par conséquent, de nombreuses zones SQL privées peuvent être associées à la même zone SQL partagée. L'emplacement d'une zone SQL privée dépend du type de connexion établi pour une session. Si une session est connectée par l'intermédiaire d'un serveur dédié, les zones SQL privées se trouvent dans la mémoire PGA du processus serveur. En revanche, si une session est connectée par l'intermédiaire d'un serveur partagé, une partie de la zone SQL privée est conservée dans la mémoire SGA.

Mémoire PGA (suite)

Zone des curseurs et zone de partage des ordres SQL

Le développeur d'un programme Oracle Pro*C ou OCI (Oracle Call Interface) peut ouvrir explicitement des *curseurs* ou des pointeurs vers des zones SQL privées spécifiques, puis les utiliser en tant que ressources nommées lors de l'exécution du programme. Les curseurs récursifs générés implicitement par la base de données pour certaines instructions SQL utilisent également des zones SQL partagées.

Zone de travail

Pour les interrogations complexes (par exemple, les interrogations d'aide à la décision), une partie importante de la mémoire PGA est dédiée aux zones de travail allouées par les opérateurs utilisant beaucoup de mémoire, tels que :

- Les opérateurs basés sur le tri, tels que ORDER BY, GROUP BY, ROLLUP et les fonctions glissantes (windowing)
- Les jointures de hachage (hash joins)
- Les fusions d'index bitmap
- La création d'index bitmap
- Les mémoires tampons d'écriture utilisées par les opérations de chargement en masse

Un opérateur de tri utilise une zone de travail (la zone de tri) pour procéder au tri d'un ensemble de lignes en mémoire. De la même façon, un opérateur de jointure de hachage utilise une zone de travail (la zone de hachage) pour créer une table de hachage à partir de son entrée gauche.

La taille d'une zone de travail peut être contrôlée et réglée. En général, l'augmentation de la taille des zones de travail permet d'améliorer de manière significative les performances d'un opérateur particulier, au détriment d'une consommation de mémoire plus élevée.

Mémoire allouée par session

La mémoire allouée par session est la mémoire allouée pour contenir les variables d'une session (informations de connexion) et d'autres informations liées à la session. Pour un serveur partagé, la mémoire allouée par session est partagée et non privée.

Gestion automatique de la mémoire PGA

Par défaut, Oracle Database gère automatiquement et de façon globale la quantité totale de mémoire dédiée à la mémoire PGA de l'instance. Vous pouvez contrôler cette quantité en définissant le paramètre d'initialisation PGA_AGGREGATE_TARGET. Oracle Database essaye alors de s'assurer que la quantité totale de mémoire PGA allouée à tous les processus serveur de base de données et processus en arrière-plan ne dépasse jamais cette cible.

Utiliser la vue V\$PARAMETER

```
SGA_TARGET = 8G
```

```
DB_CACHE_SIZE = 0  
JAVA_POOL_SIZE = 0  
LARGE_POOL_SIZE = 0  
SHARED_POOL_SIZE = 0  
STREAMS_POOL_SIZE = 0
```

```
SELECT name, value, isdefault  
FROM v$parameter  
WHERE name LIKE '%size';
```

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Utiliser la vue V\$PARAMETER

Lorsque vous affectez une valeur différente de zéro au paramètre SGA_TARGET et que vous n'indiquez aucune valeur pour les paramètres SGA à réglage automatique, la valeur de ces derniers est de 0 dans la vue V\$PARAMETER et la valeur de la colonne ISDEFAULT est TRUE.

Si vous avez défini une valeur pour un paramètre SGA à réglage automatique, la valeur qui s'affiche lorsque vous interrogez la vue V\$PARAMETER est celle que vous avez indiquée pour le paramètre en question.

Quiz

Pour obtenir des performances optimales, vous devez activer les fonctions Automatic Memory Management (AMM) et Automatic Shared Memory Management (ASMM) en définissant les paramètres `MEMORY_TARGET` et `SGA_TARGET`.

1. Vrai
2. Faux

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Réponse : 2

Synthèse

Ce chapitre vous a permis d'apprendre à :

- décrire les composants de la mémoire SGA
- implémenter la gestion automatique de la mémoire (ASMM)
- configurer manuellement les paramètres de la mémoire SGA
- utiliser la gestion automatique de la mémoire PGA

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Présentation de l'exercice 13 : Utiliser la fonction AMM pour corriger un problème d'allocation de mémoire

Cet exercice porte sur les points suivants :

- Diagnostiquer un problème d'allocation de mémoire
- Activer et implémenter la gestion automatique de la mémoire

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Unauthorized reproduction or distribution prohibited. Copyright© 2014, Oracle and/or its affiliates.

Laurent PERPETUE (laurent.perpetue@srr.fr) has a non-transferable license to use this Student Guide.

Gérer les performances de la base de données

14

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Laurent PERPETUE (laurent.perpetue@stg.fr) has a non-transferable license to use this Student Guide

Objectifs

A la fin de ce chapitre, vous pourrez :

- surveiller les performances des sessions et services
- décrire les avantages de la fonctionnalité Database Replay

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Activités de réglage

Il existe trois activités de gestion des performances :

- Planification des performances
- Réglage de l'instance
- Réglage des instructions SQL



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Activités de réglage

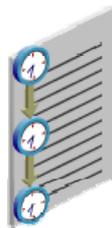
Le réglage (tuning) comprend trois facettes : planification des performances, réglage des instances et réglage des instructions SQL.

- La planification des performances est le processus qui permet de définir l'environnement : matériel, logiciel, système d'exploitation, infrastructure réseau, etc.
- Le réglage de l'instance est l'ajustement réel des paramètres de la base de données Oracle et des paramètres du système d'exploitation permettant d'améliorer les performances de la base.
- Le réglage des instructions SQL permet de faire en sorte que l'application exécute des instructions SQL efficaces. Il est effectué de façon globale pour l'application, mais aussi pour différentes instructions. Au niveau application, vous souhaitez obtenir que les différentes parties de l'application tirent parti du travail de chacune et n'entrent pas inutilement en conflit pour l'utilisation des ressources. Ce chapitre vous présente certaines actions courantes que vous pouvez effectuer pour régler des instructions SQL spécifiques.

Remarque : Pour plus d'informations sur le réglage des performances, reportez-vous au manuel *Oracle Database Performance Tuning Guide*.

Planification des performances

- Options relatives aux investissements
- Architecture du système
- Evolutivité
- Principes de conception des applications
- Test, modélisation et implémentation de la charge globale
- Déploiement de nouvelles applications



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Planification des performances

La planification des performances comporte de nombreuses facettes. Elle doit assurer un équilibre entre performances (vitesse), coût et fiabilité. Vous devez tout d'abord considérer l'investissement que vous souhaitez consacrer à l'architecture du système, c'est-à-dire à l'infrastructure matérielle et logicielle nécessaire pour répondre à vos besoins.

Bien entendu, il est nécessaire d'effectuer une analyse pour déterminer la valeur de l'environnement, de l'application et des exigences de performances concernés. Par exemple, le nombre de disques durs et de contrôleurs a un impact sur la vitesse d'accès aux données.

La capacité d'évolution d'une application est également importante. Vous devez être en mesure de pouvoir gérer un nombre croissant d'utilisateurs, de clients, de sessions ou de transactions sans que cela affecte de façon significative les performances globales du système. L'obstacle le plus important à l'évolutivité est la sérialisation des opérations entre les utilisateurs. Si tous les utilisateurs suivent le même chemin, l'un après l'autre, plus le nombre d'utilisateurs augmente, plus l'impact sur les performances se fait ressentir. Cela est dû au fait qu'un nombre croissant d'utilisateurs attend de pouvoir accéder à ce chemin.

Les instructions SQL non optimisées affectent également l'évolutivité. Elles obligent de nombreux utilisateurs à attendre la fin de leur exécution. Ces derniers sont en rivalité les uns avec les autres pour l'utilisation d'un grand nombre de ressources dont ils n'ont en réalité pas besoin.

Les principes de conception des applications peuvent aussi avoir un impact significatif sur les performances. Une conception simple, l'utilisation de vues et d'index et la modélisation des données sont autant d'éléments essentiels.

Planification des performances (suite)

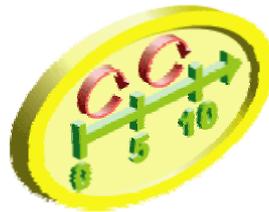
Toute application doit être testée avec une charge globale de production représentative. Cela implique d'estimer la taille de la base et la charge globale, puis de générer des données de test et une charge système.

Les performances doivent être reconsidérées au fur et à mesure que de nouvelles applications (ou de nouvelles versions d'application) sont déployées. Certaines décisions de conception sont parfois prises pour maintenir la compatibilité avec d'anciens systèmes lors du déploiement. Une nouvelle base de données doit être spécialement configurée pour les applications hébergées (en fonction de l'environnement de production).

Il est difficile mais nécessaire de tester les applications existantes lors de la modification de l'infrastructure. C'est le cas, par exemple, en cas de mise à niveau de la base de données vers une version plus récente, ou de changement de système d'exploitation ou de matériel serveur. Avant le déploiement de l'application pour production dans la nouvelle configuration, vous souhaitez en connaître l'impact. Il est fort probable que l'application aura besoin de réglages supplémentaires. Vous devez être sûr que les fonctions critiques s'exécuteront sans erreur.

Réglage de l'instance

- Les objectifs doivent être bien définis.
- Une mémoire suffisante doit être allouée aux structures de la base de données.
- Les besoins en matière d'E/S doivent être considérés dans chaque partie de la base de données.
- Le système d'exploitation doit être réglé pour que les performances de la base de données soient optimales.



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Réglage de l'instance

Au début de toute activité de réglage (tuning), il est nécessaire de définir des objectifs précis. Par exemple, il est plus facile de travailler avec un objectif du type "Traiter 500 transactions commerciales par minute" qu'avec un objectif tel que "Obtenir le débit le plus élevé possible et on verra".

Vous devez allouer à la base de données Oracle une quantité de mémoire suffisante pour que les performances de l'application soient optimales. Toutefois, la quantité de mémoire disponible est limitée. L'allocation d'une quantité insuffisante de mémoire à certaines parties de la base Oracle peut entraîner une activité en arrière-plan inefficace, laquelle peut passer inaperçue en l'absence d'analyse.

Les E/S disque constituent souvent un goulet d'étranglement. Elles nécessitent par conséquent une attention particulière au début de toute implémentation de base de données.

La configuration du système d'exploitation peut également affecter les performances d'une base Oracle. Pour plus d'informations, reportez-vous au manuel *Oracle Database Installation Guide* correspondant à la plate-forme que vous utilisez.

Méthodologie de réglage des performances

Pour effectuer un réglage, respectez les règles suivantes :

- Procédez à un réglage descendant. Respectez l'ordre suivant :
 1. Conception
 2. Code de l'application
 3. Instance
- Réglez la zone qui revêt l'intérêt potentiel le plus important. Identifiez et réglez les éléments suivants :
 - Instructions SQL utilisant le plus de ressources
 - Attentes les plus longues
 - Durées de service les plus longues
- Arrêtez le réglage lorsque l'objectif est atteint.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Méthodologie de réglage des performances

La méthodologie de réglage (tuning) développée par Oracle repose sur des années d'expérience. Les étapes de base sont les suivantes :

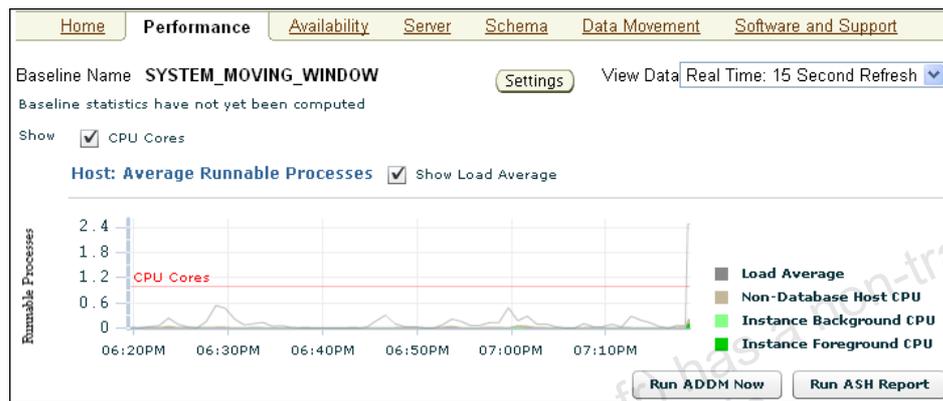
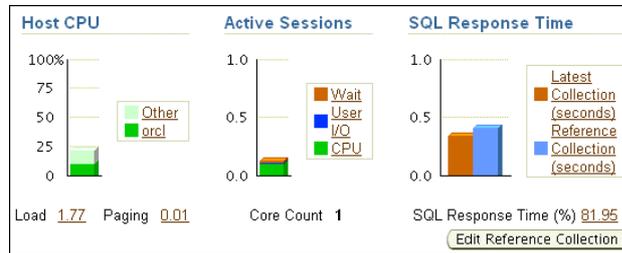
1. Vérifiez les statistiques du système d'exploitation et l'état général de l'ordinateur avant de régler l'instance pour être sûr que le problème vient de la base de données. Dans Enterprise Manager, utilisez la page d'accueil de la base.
2. Procédez à un réglage descendant. Réglez la conception, puis l'application, et enfin l'instance. Par exemple, avant de régler la présentation des tablespaces sur le disque, essayez d'éliminer les balayages complets de table (full table scans) provoquant des contentions au niveau des E/S. Cette activité requiert souvent l'accès au code de l'application.
3. Réglez la zone qui revêt l'intérêt potentiel le plus important. La méthodologie de réglage présentée dans ce cours est simple. Identifiez le goulet d'étranglement le plus important et réglez-le. Répétez cette étape. Les différents outils de réglage permettent d'une façon ou d'une autre d'identifier les instructions SQL, les contentions de ressources ou les services qui consomment le plus de temps. La base de données Oracle fournit un modèle et des mesures de performances spécifiques qui permettent d'automatiser le processus d'identification des goulets d'étranglement. Les fonctions de conseil disponibles dans Oracle Database 11g utilisent précisément cette méthodologie.
4. Arrêtez le réglage lorsque votre objectif est atteint. Cette étape implique de définir les objectifs du réglage.

Le processus général de réglage d'une instance de base de données présenté ici peut nécessiter plusieurs passages.

Surveillance des performances

Avec Enterprise Manager :

- Aperçu des performances
- Graphiques des mesures et des données détaillées



Copyright © 2009, Oracle. Tous droits réservés.

Surveillance des performances

Vous ne pouvez répondre à une baisse des performances que si vous en êtes informé. C'est pourquoi Oracle Database 11g vous permet de surveiller de différentes manières les performances actuelles de votre instance de base de données. Dans Enterprise Manager (EM), la page d'accueil de la base de données fournit un bref aperçu de l'état général de l'instance et du serveur, ainsi que des graphiques illustrant l'utilisation de la CPU, les sessions actives et le temps de réponse des instructions SQL. Elle présente également les alertes qui ont été déclenchées.

L'onglet Performance d'EM présente plusieurs graphiques de mesures des performances selon divers points de vue. Vous pouvez voir les performances en termes de CPU, de nombre moyen de sessions actives, de débit, d'E/S et autres. Dans la page Performance, vous pouvez cliquer sur des liens permettant d'accéder à des informations détaillées relatives aux sessions et aux différentes instructions SQL.

Les informations affichées dans EM sont fondées sur des vues de performances figurant dans la base de données. Vous pouvez accéder à ces vues directement avec SQL*Plus. Il peut arriver que vous ayez besoin d'accéder à ces vues pour obtenir des détails sur les statistiques brutes.

Données de réglage des performances

Type de données collectées :

- Statistiques cumulées :
 - Événements Wait avec informations temporelles
 - Jeu de statistiques temporelles Time Model
- Mesures de performances : taux statistiques
- Statistiques échantillonnées : historique des sessions actives (Active Session History - ASH)
 - Statistiques par session
 - Statistiques par instruction SQL
 - Statistiques par service
 - Autres



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Données de réglage des performances

Le logiciel serveur de base de données Oracle capture des informations sur son propre fonctionnement. Trois principaux types de données sont collectés : des statistiques cumulées, des mesures de performances et des statistiques échantillonnées.

Les statistiques cumulées sont des nombres et des informations temporelles relatives à différents événements qui se produisent sur le serveur de base de données.

Certains événements sont relativement importants, comme les `buffer busy waits`.

D'autres ont un faible impact sur le réglage, comme les `index block split`.

Les événements les plus importants pour le réglage sont généralement ceux qui présentent les valeurs temporelles cumulées les plus élevées. Dans Oracle Database 11g, les statistiques sont corrélées grâce à un jeu de statistiques temporelles Time Model. Ces statistiques temporelles sont fondées sur un pourcentage de temps d'exécution de la base de données qui constitue un socle de comparaison commun.

Les mesures de performances sont des statistiques par unité. L'unité peut être temporelle (des secondes, par exemple) ou correspondre à une transaction ou une session. Les mesures de performances sont le fondement de la surveillance proactive des performances.

Vous pouvez leur affecter des seuils déclenchant une alerte. Par exemple, vous pouvez définir des seuils pour déclencher une alerte lorsque les lectures par milliseconde dépassent un record précédemment enregistré ou quand la zone de stockage des fichiers de journalisation archivés est pleine à 95 %.

Des statistiques échantillonnées sont collectées automatiquement lorsque le paramètre `STATISTICS_LEVEL` a pour valeur `TYPICAL` ou `ALL`. Ces statistiques vous permettent de discerner une évolution. Vous pouvez consulter des statistiques relatives aux sessions ainsi que des statistiques système collectées précédemment, selon différentes perspectives, même si vous n'aviez pas préalablement pensé à demander la collecte de ces données.

Collecte des statistiques destinées à l'optimiseur

- Réglage des performances SQL : nécessite de disposer de statistiques exactes.
- Statistiques destinées à l'optimiseur :
 - Statistiques sur les objets
 - Statistiques relatives au système d'exploitation
- Modes de collecte des statistiques :
 - Automatiquement : Tâches de maintenance automatisées
 - Manuellement : Package DBMS_STATS
 - En configurant les paramètres d'initialisation de la base de données
 - En important des statistiques à partir d'une autre base de données

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Collecte des statistiques destinées à l'optimiseur

Les statistiques destinées à l'optimiseur sont des ensembles de données qui fournissent des détails spécifiques sur les objets de base de données. Ces statistiques sont essentielles pour permettre à l'optimiseur d'instructions de choisir le meilleur plan d'exécution pour chaque instruction SQL. Elles sont collectées de façon régulière et ne changent pas entre les collectes.

L'approche recommandée pour la collecte des statistiques destinées à l'optimiseur est de permettre à la base de données Oracle de réaliser l'opération automatiquement. Les tâches de maintenance automatisées peuvent être créées automatiquement lors de la création de la base et sont gérées par le planificateur. Elles collectent par défaut des statistiques sur tous les objets de la base de données dont des statistiques sont manquantes ou obsolètes. Vous pouvez modifier la configuration par défaut via la page Automatic Maintenance Tasks.

Les statistiques système décrivent à l'optimiseur d'instructions les caractéristiques matérielles du système, telles que les performances et l'utilisation des E/S et de la CPU. Lorsque vous choisissez un plan d'exécution, l'optimiseur évalue les ressources d'E/S et de CPU requises pour chaque interrogation. Les statistiques système permettent à l'optimiseur d'instructions d'évaluer plus précisément les coûts en termes d'E/S et de CPU, et ainsi de choisir un meilleur plan d'exécution. Les statistiques système sont collectées à l'aide de la procédure DBMS_STATS.GATHER_SYSTEM_STATS. Lorsque la base de données Oracle collecte les statistiques système, elle analyse l'activité du système sur une période donnée. Les statistiques système ne sont pas collectées automatiquement. Oracle Corporation recommande d'utiliser le package DBMS_STATS pour les recueillir.

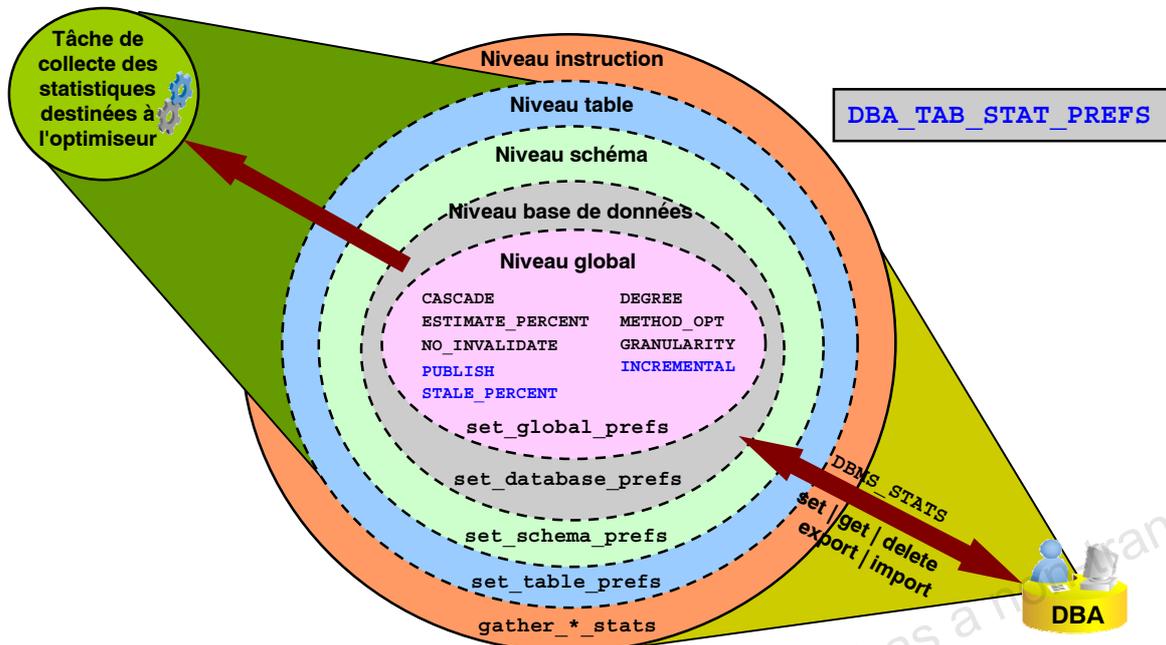
Collecte des statistiques destinées à l'optimiseur (suite)

Si vous choisissez de ne pas utiliser la collecte automatique des statistiques, vous devez collecter les statistiques manuellement dans tous les schémas, y compris les schémas système. Si les données de la base changent fréquemment, vous devez en outre collecter les statistiques régulièrement, afin de garantir qu'elles représentent de manière précise les caractéristiques des objets de base de données. Pour collecter manuellement les statistiques, utilisez le package DBMS_STATS. Ce package PL/SQL est également utilisé pour modifier, afficher, exporter, importer et supprimer des statistiques.

Vous pouvez aussi gérer la collecte des statistiques destinées à l'optimiseur et des statistiques système par l'intermédiaire de paramètres d'initialisation de la base de données. Par exemple :

- Le paramètre OPTIMIZER_DYNAMIC_SAMPLING contrôle le niveau d'échantillonnage dynamique effectué par l'optimiseur. Vous pouvez utiliser l'échantillonnage dynamique pour évaluer les statistiques relatives aux tables et aux index pertinents lorsqu'elles ne sont pas disponibles ou sont trop anciennes pour être considérées comme valides. L'échantillonnage dynamique évalue également la sélectivité des prédicats de table unique, lorsque les statistiques collectées ne peuvent pas être utilisées ou sont susceptibles de provoquer des erreurs importantes dans l'estimation.
- Le paramètre STATISTICS_LEVEL contrôle les principales collectes de statistiques ou fonctions de conseil de la base de données et définit le niveau de collecte de statistiques de la base. La valeur de ce paramètre peut être BASIC, TYPICAL ou ALL. Vous pouvez interroger la vue V\$STATISTICS_LEVEL pour déterminer les paramètres affectés par le paramètre STATISTICS_LEVEL.
Remarque : Le fait d'affecter la valeur BASIC à STATISTICS_LEVEL désactive de nombreuses fonctionnalités automatiques, ce qui n'est pas recommandé.

Préférences relatives aux statistiques : Présentation



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Préférences relatives aux statistiques : Présentation

La fonctionnalité automatisée de collecte des statistiques a été introduite dans Oracle Database 10g Release 1 pour simplifier la gestion des statistiques destinées à l'optimiseur. Dans certains cas, toutefois, vous deviez la désactiver et exécuter vos propres scripts. Par exemple, cela pouvait se produire à cause du manque de contrôle au niveau objet. Chaque fois que vous étiez en présence d'un petit sous-ensemble d'objets pour lequel les options par défaut de collecte des statistiques n'étaient pas adéquates, vous deviez verrouiller les statistiques et les analyser séparément en utilisant vos propres options. Par exemple, la fonctionnalité qui tente automatiquement de déterminer la taille d'échantillon adéquate (`ESTIMATE_PERCENT=AUTO_SAMPLE_SIZE`) n'est pas performante avec des colonnes qui contiennent des données présentant des écarts de fréquence importants. Le seul moyen de contourner ce problème était d'indiquer manuellement la taille d'échantillon dans un script défini par vos soins.

Remarque : La vue `DBA_TAB_STAT_PREFS` indique tous les paramètres effectifs de préférences sur les statistiques, pour toutes les tables pertinentes.

Utiliser les préférences relatives aux statistiques

- **PUBLISH** : Permet de publier les statistiques dans le dictionnaire ou de les stocker préalablement dans une zone d'attente.
- **STALE_PERCENT** : Permet de déterminer le seuil à partir duquel un objet est considéré comme contenant des statistiques obsolètes. La valeur correspond au pourcentage de lignes modifiées depuis la dernière collecte de statistiques.
- **INCREMENTAL** : Permet de collecter les statistiques globales sur des tables partitionnées par incrémentation.

```
exec dbms_stats.set_table_prefs('SH','SALES','STALE_PERCENT','13');
```

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Utiliser les préférences relatives aux statistiques

Les préférences relatives aux statistiques d'Oracle Database 11g vous permettent de vous appuyer davantage sur la fonctionnalité de collecte automatique pour gérer les statistiques de l'optimiseur lorsque certains objets nécessitent des paramètres différents des paramètres par défaut de la base de données.

Cette fonctionnalité vous permet d'associer les options de collecte des statistiques qui modifient le comportement par défaut des procédures `GATHER_*_STATS` et la tâche automatique de collecte des statistiques destinées à l'optimiseur au niveau objet ou schéma. Vous pouvez utiliser le package `DBMS_STATS` pour gérer les options de collecte des statistiques.

Les préférences peuvent être définies, extraites, supprimées, exportées et importées à chaque niveau : table, schéma, base de données, global. Les préférences globales sont utilisées pour les tables sans préférences, alors que les préférences de base de données sont utilisées pour définir des préférences sur toutes les tables.

Les options suivantes sont nouvelles dans Oracle Database 11g Release 1 :

- **PUBLISH** vous permet de publier les statistiques dans le dictionnaire ou de les stocker préalablement dans une zone d'attente.
- **STALE_PERCENT** permet de déterminer le seuil à partir duquel les statistiques relatives à un objet sont obsolètes. La valeur correspond au pourcentage de lignes modifiées depuis la dernière collecte de statistiques. L'exemple présenté dans la diapositive remplace la valeur par défaut (10 %) par 13 % pour `SH.SALES`.
- **INCREMENTAL** permet de collecter des statistiques globales sur des tables partitionnées, de manière incrémentielle.

Configurer les préférences globales avec Enterprise Manager

The screenshot illustrates the navigation path in Oracle Enterprise Manager 11g to configure global statistics gathering preferences. It shows the 'Database Instance: orcl' page with the 'Server' tab selected. From there, the user navigates to 'Manage Optimizer Statistics' and then to 'Global Statistics Gathering Options'. The 'Global Statistics Gathering Options' page displays the 'Gather Optimizer Statistics Default Options' section, which includes various configuration options such as Estimate Percentage, Degree of Parallelism, Granularity, Cursor Invalidation, Cascade, Target Object Class, Stale Percentage, Incremental, Publish, and Histograms.

Copyright © 2009, Oracle. Tous droits réservés.

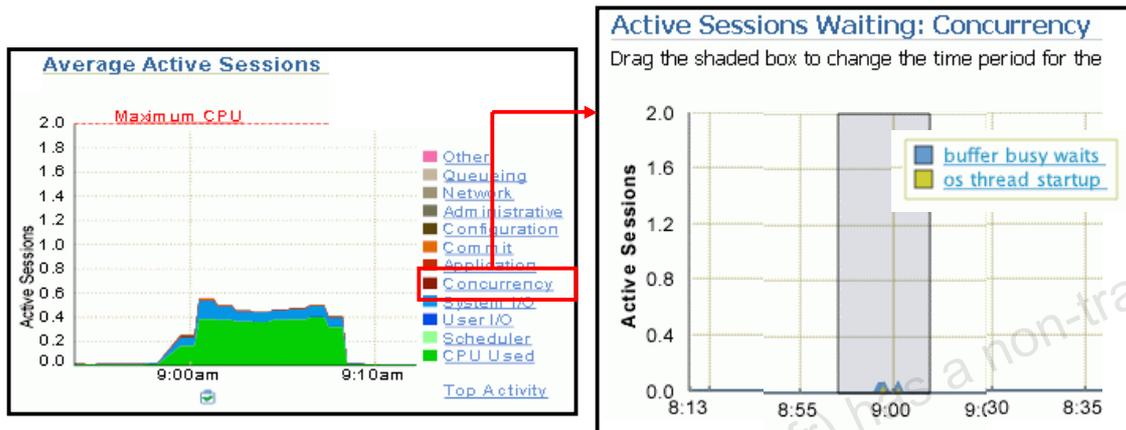
Configurer les préférences globales avec Enterprise Manager

Vous pouvez utiliser Enterprise Manager pour contrôler les paramètres relatifs aux préférences globales. Pour cela, utilisez la page Manage Optimizer Statistics, à laquelle vous accédez à partir de la page d'accueil Database en cliquant sur l'onglet Server, puis sur le lien Manage Optimizer Statistics, et enfin sur le lien Global Statistics Gathering Options. Dans la page Global Statistics Gathering Options, modifiez les préférences globales dans la section Gather Optimizer Statistics Default Options. Ensuite, cliquez sur le bouton Apply.

Remarque : Pour modifier les options de collecte des statistiques au niveau objet ou schéma, cliquez sur le lien Object Level Statistics Gathering Preferences dans la page Manage Optimizer Statistics.

Événements Wait Oracle

- Un ensemble d'événements Wait fournit des informations sur les sessions ou processus qui ont dû ou qui doivent attendre pour différentes raisons.
- Ces événements sont répertoriés dans la vue `V$EVENT_NAME`.



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Événements Wait Oracle

Les événements Wait sont des statistiques qui sont incrémentées par un processus ou thread serveur afin d'indiquer qu'il a dû attendre la fin d'un événement pour poursuivre le traitement. Les données des événements Wait révèlent les symptômes de divers problèmes pouvant avoir un impact sur les performances, par exemple la contention au niveau d'un verrou interne (latch), de la mémoire tampon (buffer) ou des E/S. Rappelez-vous qu'il ne s'agit là que de symptômes, et non des causes proprement dites.

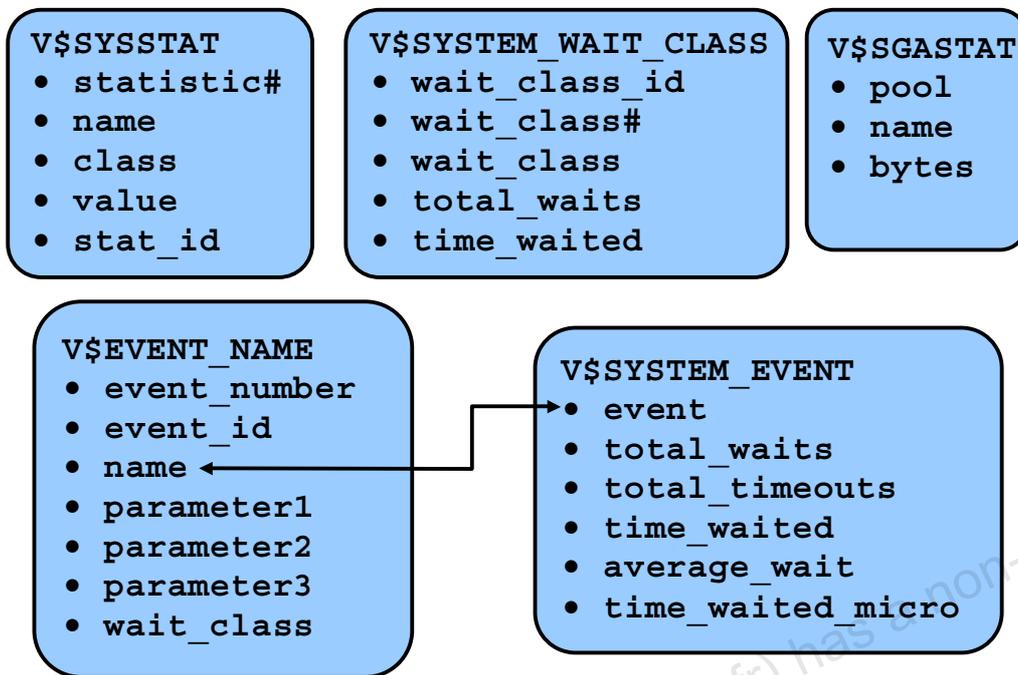
Les événements Wait sont regroupés en classes. Ces classes sont les suivantes : Administrative, Application, Cluster, Commit, Concurrency, Configuration, Idle, Network, Other, Scheduler, System I/O et User I/O.

Il existe plus de 800 événements Wait dans la base de données Oracle, notamment `free buffer wait`, `latch free`, `buffer busy waits`, `db file sequential read` et `db file scattered read`.

A partir d'Enterprise Manager, vous pouvez afficher les événements Wait en ouvrant la page Performance, puis en affichant le graphique Average Active Sessions, comme le montre la diapositive ci-dessus. En cliquant sur le lien d'une classe d'événement Wait particulière, vous pouvez procéder à une hiérarchisation descendante vers des événements Wait spécifiques via l'interface Top Activity. Dans l'exemple de la diapositive ci-dessus, le nombre de `buffer busy waits` était très faible.

Pour obtenir la liste des événements Oracle les plus courants, reportez-vous à la documentation *Oracle Database Reference 11g*.

Statistiques au niveau instance



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Statistiques au niveau instance

Pour un diagnostic efficace des problèmes de performances, des statistiques doivent être disponibles. L'instance de base de données Oracle génère de nombreux types de statistiques cumulées pour le système, les sessions et les instructions SQL individuelles. Par ailleurs, elle effectue le suivi des statistiques cumulées sur les segments et les services.

Lorsque vous analysez un problème de performances dans l'un de ces domaines, vous examinez généralement les modifications des statistiques (valeur delta) sur la période qui vous intéresse.

Remarque : Les statistiques au niveau instance sont dynamiques et sont initialisées à chaque démarrage d'instance. Il est possible de les capturer à un moment spécifique et de les conserver dans la base sous forme de clichés.

Statistiques sur les événements Wait

Tous les événements possibles sont catalogués dans la vue V\$EVENT_NAME.

Les statistiques cumulées pour toutes les sessions sont stockées dans la vue V\$SYSTEM_EVENT, qui affiche le total des attentes pour un événement particulier depuis le démarrage de l'instance.

Lorsque vous procédez à la résolution des problèmes, vous devez savoir si un processus a attendu des ressources.

Statistiques au niveau instance (suite)

Statistiques à l'échelle du système

Toutes les statistiques à l'échelle du système sont cataloguées dans la vue V\$STATNAME : plus de 400 statistiques sont disponibles dans Oracle Database 11g.

Le serveur affiche dans la vue V\$SYSSTAT toutes les statistiques système calculées.

Vous pouvez interroger cette vue afin de déterminer les totaux cumulés depuis le démarrage de l'instance.

Exemple

```
SQL> SELECT name, class, value FROM v$sysstat;
NAME                                CLASS          VALUE
-----
...
table scans (short tables)          64             135116
table scans (long tables)           64              250
table scans (rowid ranges)          64              0
table scans (cache partitions)      64              3
table scans (direct read)           64              0
table scan rows gotten              64          14789836
table scan blocks gotten            64           558542
...
```

Les statistiques à l'échelle du système sont organisées en fonction du sujet de réglage et des objectifs de débogage. Les classes incluent généralement l'activité générale des instances, l'activité des tampons de journalisation (redo log buffers), le verrouillage, l'activité du cache de tampons (buffer cache) de la base de données, etc. Etant donné que les différentes statistiques système peuvent appartenir à plusieurs classes, il n'est pas possible d'effectuer une jointure simple entre V\$SYSSTATS.CLASS et V\$SYSTEM_WAIT_CLASS.WAIT_CLASS#.

Vous pouvez afficher tous les événements Wait d'une classe particulière en interrogeant la vue V\$SYSTEM_WAIT_CLASS, comme dans l'exemple suivant (avec application de la mise en forme) :

```
SQL> SELECT * FROM V$SYSTEM_WAIT_CLASS
2 WHERE wait_class LIKE '%I/O%';
CLASS_ID  CLASS#  WAIT_CLASS  TOTAL_WAITS  TIME_WAITED
-----
1740759767    8 User I/O    1119152      39038
4108307767    9 System I/O  296959       27929
```

Statistiques au niveau instance (suite)

Statistiques globales concernant la mémoire SGA

Le serveur affiche dans la vue V\$SGASTAT toutes les statistiques mémoire calculées. Vous pouvez interroger cette vue afin de déterminer les totaux cumulés de l'utilisation détaillée de la mémoire SGA depuis le démarrage de l'instance.

```
SQL> SELECT * FROM v$sgastat;
POOL          NAME                                BYTES
-----
              fixed_sga                    7780360
              buffer_cache                 25165824
              log_buffer                   262144
shared pool   sessions                          1284644
shared pool   sql area                       22376876
...
```

Les résultats illustrés ci-dessus ne représentent qu'une partie de la sortie.

Lorsque la valeur du paramètre STATISTICS_LEVEL est BASIC, celle du paramètre TIMED_STATISTICS est, par défaut, FALSE. Aucune information temporelle n'est collectée pour les événements Wait et la plupart des fonctionnalités de surveillance des performances de la base de données sont désactivées. Le paramétrage explicite du paramètre TIMED_STATISTICS remplace la valeur dérivée du paramètre STATISTICS_LEVEL.

Surveiller les performances des sessions



ORACLE

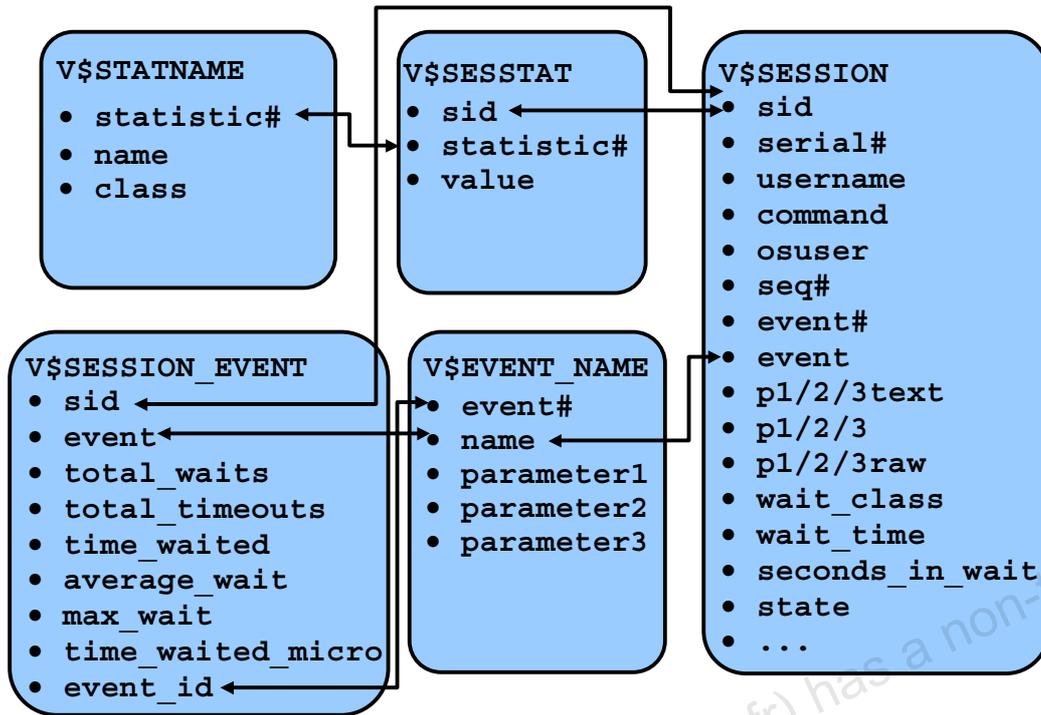
Copyright © 2009, Oracle. Tous droits réservés.

Surveiller les performances des sessions

Certaines pages d'Enterprise Manager sont consacrées aux détails des sessions et présentent les événements Wait qui se produisent au cours de sessions particulières. Dans l'onglet Performance, cliquez sur Top Activity pour consulter un récapitulatif de l'ensemble des sessions. Dans l'angle inférieur droit de la page Top Activity figure la liste des sessions les plus consommatrices de ressources. Cliquez sur l'identificateur d'une session pour afficher la page Session Details.

Les pages Top Activity et Session Details sont fondées sur les vues des performances de la base de données.

Afficher les statistiques liées aux sessions



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Afficher les statistiques liées aux sessions

Vous pouvez afficher les informations relatives aux sessions actuelles pour chaque utilisateur connecté en interrogeant la vue V\$SESSION. Par exemple, vous pouvez utiliser la vue V\$SESSION pour déterminer si une session représente une session utilisateur ou a été créée par un processus serveur de base de données (BACKGROUND).

Vous pouvez interroger la vue V\$SESSION ou V\$SESSION_WAIT afin de déterminer les ressources ou événements que les sessions actives attendent.

Vous pouvez consulter des statistiques relatives aux sessions utilisateur dans la vue V\$SESSTAT. La vue V\$SESSION_EVENT affiche les informations relatives aux attentes d'une session pour un événement donné.

Les valeurs cumulées des statistiques au niveau instance sont généralement disponibles par l'intermédiaire de vues dynamiques des performances, telles que les vues V\$SESSTAT et V\$SYSSTAT. Notez que les valeurs cumulées dans les vues dynamiques sont réinitialisées lorsque l'instance de base de données est arrêtée.

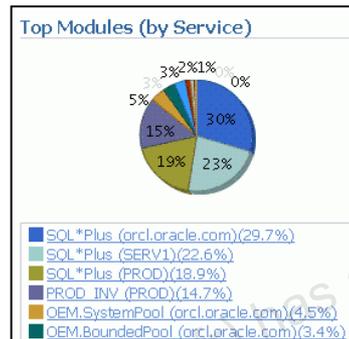
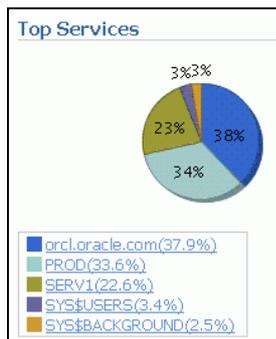
La vue V\$MYSTAT affiche les statistiques de la session en cours.

Vous pouvez également interroger la vue V\$SESSMETRIC afin d'afficher les valeurs de performances pour toutes les sessions actives. Cette vue répertorie des mesures de performances telles que l'utilisation de la CPU, le nombre de lectures physiques, le nombre d'analyses complètes (hard parses) et le taux de lecture logique.

Afficher les statistiques liées aux services

Pour les environnements à n niveaux (n -tiers), les statistiques liées aux sessions ne sont pas très utiles. Vous pouvez afficher les statistiques de niveau service dans les vues suivantes :

- **V\$SERVICE_EVENT** : Cumul du nombre d'attentes et du temps d'attente pour chaque service, par événement
- **V\$SERVICE_WAIT_CLASS** : Cumul du nombre d'attentes et du temps d'attente pour chaque service, par classe wait



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Afficher les statistiques liées aux services

Dans un environnement à n niveaux (n -tiers), dans lequel un serveur d'applications regroupe les connexions de base de données, il se peut que l'affichage des sessions ne fournisse pas les informations souhaitées pour analyser les performances. Le regroupement de sessions au sein d'un service vous permet de surveiller les performances de façon plus précise. Les deux vues mentionnées dans la diapositive ci-dessus fournissent les mêmes informations que leurs équivalents relatifs aux sessions, à ceci près que les informations sont présentées au niveau service et non au niveau session.

V\$SERVICE_WAIT_CLASS affiche les statistiques d'attente de chaque service, par classe wait.

V\$SERVICE_EVENT affiche les mêmes informations que V\$SERVICE_WAIT_CLASS, réparties par ID d'événement.

Enterprise Manager présente également un cumul par service, d'une part, et par module et service, d'autre part. Dans chacune des vues, vous pouvez cliquer sur la légende afin de consulter l'activité et les statistiques de chaque service.

Vous pouvez définir un service dans la base de données à l'aide du package DBMS_SERVICE et utiliser le nom de service réseau pour affecter des applications à un service.

Vues de dépannage et de réglage

Instance/Base de données

V\$DATABASE
V\$INSTANCE
V\$PARAMETER
V\$SPPARAMETER
V\$SYSTEM_PARAMETER
V\$PROCESS
V\$BGPROCESS
V\$PX_PROCESS_SYSSTAT
V\$SYSTEM_EVENT

Disque

V\$DATAFILE
V\$FILESTAT
V\$LOG
V\$LOG_HISTORY
V\$DBFILE
V\$TEMPFILE
V\$TEMPSEG_USAGE
V\$SEGMENT_STATISTICS

Mémoire

V\$BUFFER_POOL_STATISTICS
V\$LIBRARYCACHE
V\$SGAINFO
V\$PGASTAT

Contention

V\$LOCK
V\$UNDOSTAT
V\$WAITSTAT
V\$LATCH

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Vues de dépannage et de réglage

La diapositive ci-dessus répertorie quelques-unes des vues auxquelles vous pouvez avoir besoin d'accéder pour déterminer la cause des problèmes de performances ou pour analyser le statut actuel de la base. Beaucoup de ces vues présentent les mêmes données que celles utilisées pour la création des états générés par Enterprise Manager à partir du référentiel AWR (Automatic Workload Repository). Dans certains cas, les données brutes sont nécessaires pour un diagnostic complet du problème.

Pour une description complète de ces vues, reportez-vous au manuel *Oracle Database Reference Manual*.

Vues du dictionnaire

- Les vues du dictionnaire et vues spéciales suivantes fournissent des statistiques au niveau objet après l'utilisation du package DBMS_STATS :
 - DBA_TABLES, DBA_TAB_COLUMNS
 - DBA_CLUSTERS
 - DBA_INDEXES
 - DBA_TAB_HISTOGRAMS
- Ces informations statistiques sont statiques jusqu'à ce que vous réexécutez les procédures appropriées dans DBMS_STATS.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Vues du dictionnaire

Lorsque vous devez examiner en détail les statistiques relatives à des objets de base de données spécifiques, utilisez le package DBMS_STATS, lequel collecte des statistiques et alimente les colonnes de certaines vues DBA_XXX.

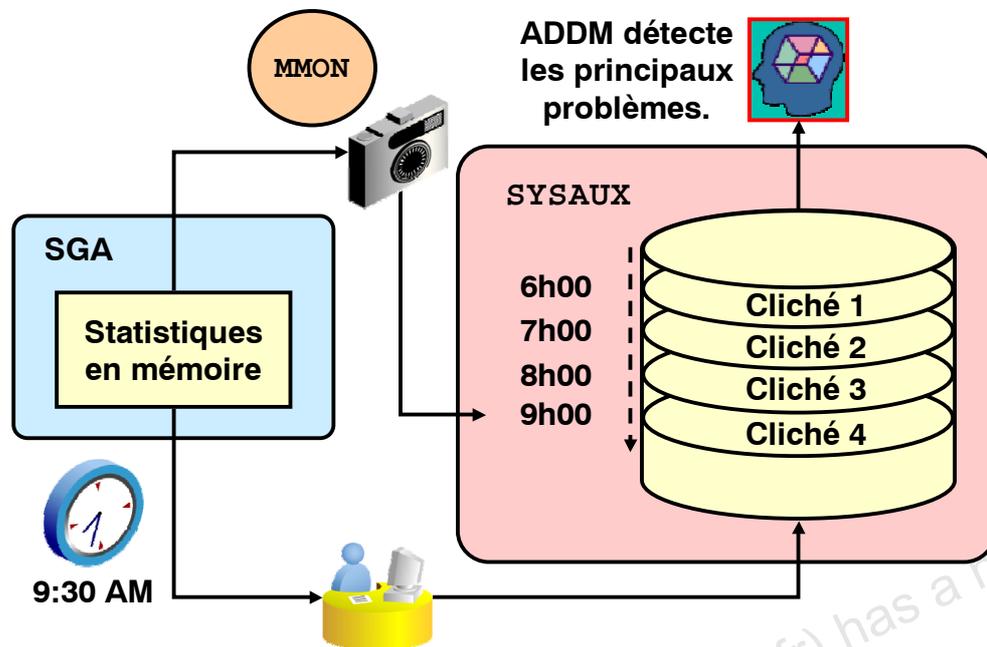
Le package DBMS_STATS alimente les colonnes des vues concernant :

- Le stockage des données des tables dans les extents (ensembles de blocs contigus) et les blocs :
 - DBA_TABLES
 - DBA_TAB_COLUMNS
- Le stockage des données des clusters dans les extents et les blocs :
 - DBA_CLUSTERS
- Le stockage des données d'index dans les extents et les blocs, ainsi que l'utilité des index :
 - DBA_INDEXES
- La distribution des données dans les colonnes non indexées et indexées :
 - DBA_TAB_HISTOGRAMS

Pour plus d'informations sur l'utilisation du package DBMS_STATS, reportez-vous au manuel *Oracle Database Performance Tuning Guide*.

La commande ANALYZE INDEX ... VALIDATE STRUCTURE permet d'alimenter les vues INDEX_STATS et INDEX_HISTOGRAM qui contiennent des statistiques relatives aux index.

Référentiel AWR



Copyright © 2009, Oracle. Tous droits réservés.

Référentiel AWR

Le référentiel AWR (Automatic Workload Repository) est un ensemble de statistiques de performances système persistantes, appartenant à SYS. Ce référentiel réside dans le tablespace SYSAUX.

Un *cliché* est un ensemble de statistiques de performances, capturées à un instant donné et stockées dans le référentiel AWR. Chaque cliché est identifié par un numéro de séquence (`snap_id`), unique dans le référentiel AWR. Par défaut, les clichés sont générés toutes les 60 minutes. Vous pouvez ajuster cette fréquence en modifiant le paramètre de cliché `INTERVAL`. Etant donné que les fonctions de conseil (advisors) de la base de données se basent sur ces clichés, n'oubliez pas que l'ajustement de l'intervalle peut affecter la précision des diagnostics. Par exemple, si une valeur de 4 heures est affectée au paramètre `INTERVAL`, vous risquez de manquer les événements transitoires qui sont perceptibles pendant les intervalles de 60 minutes.

Vous pouvez utiliser la procédure stockée `DBMS_WORKLOAD_REPOSITORY.MODIFY_SNAPSHOT_SETTINGS` ou Database Control pour modifier les paramètres qui contrôlent la collecte des clichés (snapshots). Dans Database Control, cliquez sur Automatic Workload Repository dans la région Statistics Management de l'onglet Server. Cliquez ensuite sur Edit pour apporter les modifications. La procédure stockée offre davantage de flexibilité que Database Control pour la définition des valeurs `INTERVAL`.

Référentiel AWR (suite)

Vous pouvez prendre des clichés manuels à l'aide de Database Control ou de la procédure stockée `DBMS_WORKLOAD_REPOSITORY.CREATE_SNAPSHOT`. La prise de clichés manuels est effectuée conjointement avec les clichés automatiques générés par le système. Les clichés manuels peuvent être utilisés lorsque vous souhaitez capturer le comportement du système à deux instants spécifiques qui ne coïncident pas avec la planification automatique.

Statspack est un utilitaire qui fournit un sous-ensemble des fonctionnalités de collecte et de création d'états du référentiel AWR. Il n'existe cependant aucun moyen de migration des données Statspack vers le référentiel de charge globale. En outre, le référentiel de charge globale n'est pas compatible avec le schéma Statspack. Par ailleurs, Statspack n'est pas accessible via Enterprise Manager. Il requiert une configuration particulière et ne présente pas de paramètres de conservation automatique, ni de purge automatique. L'utilitaire Statspack fournit des scripts de configuration, de collecte automatique de clichés (snapshots) et de création d'états. Les clichés Statspack peuvent être marqués pour conservation, dans le cadre d'une ligne de base (baseline) Statspack, ou purgés avec les scripts fournis.

Statspack est documenté dans le fichier `$ORACLE_HOME/rdbms/admin/spdoc.txt`.

Utiliser les vues du référentiel AWR

- DBA_HIST_DB_CACHE_ADVICE
- DBA_HIST_DISPATCHER
- DBA_HIST_DYN_REMASTER_STATS
- DBA_HIST_IOSTAT_DETAIL
- DBA_HIST_SHARED_SERVER_SUMMARY

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Utiliser les vues du référentiel AWR

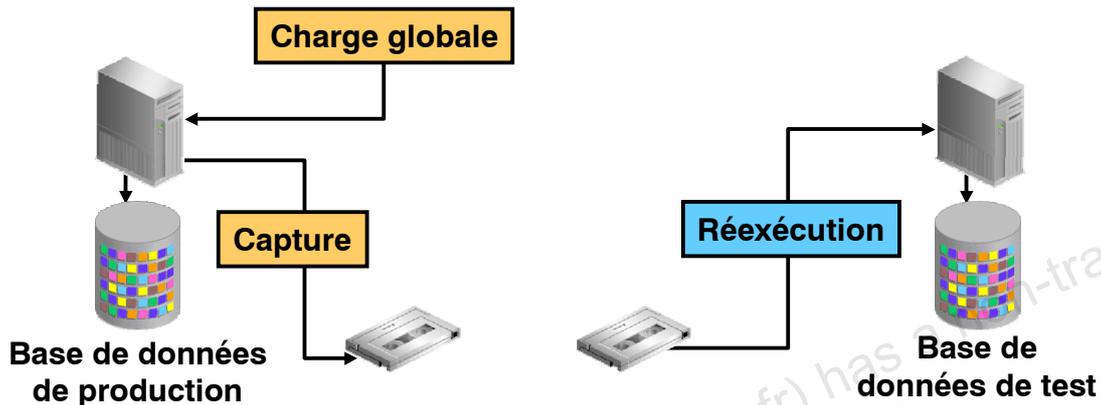
Les nouvelles vues de statistiques AWR suivantes sont disponibles dans Oracle Database 11g Release 2 :

- DBA_HIST_DB_CACHE_ADVICE : Affiche les prévisions historiques concernant le nombre de lectures physiques pour la taille de cache correspondant à chaque ligne.
- DBA_HIST_DISPATCHER : Affiche les informations historiques de chaque processus répartiteur au moment du cliché (snapshot).
- DBA_HIST_DYN_REMASTER_STATS : Affiche les informations statistiques sur le processus de redistribution dynamique.
- DBA_HIST_IOSTAT_DETAIL : Affiche les statistiques d'E/S historiques regroupées par type de fichier et par fonction.
- DBA_HIST_SHARED_SERVER_SUMMARY : Affiche les informations historiques pour les serveurs partagés, telles que l'activité du serveur partagé, les files d'attente communes et les files d'attente du répartiteur.

Présentation de Real Application Testing : Database Replay

Database Replay :

- capture les charges globales de production
- effectue des tests avec des charges globales réalistes
- réexécute la même instruction SQL sur les mêmes données à chaque test



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Présentation de Real Application Testing : Database Replay

Un changement va être apporté au niveau du système : une mise à niveau du matériel, du système d'exploitation ou de la version de la base de données (voire des trois) doit être effectuée. Vous devez assurer aux utilisateurs que l'application continuera à fonctionner après la mise à niveau et que les performances seront au moins aussi bonnes qu'avant.

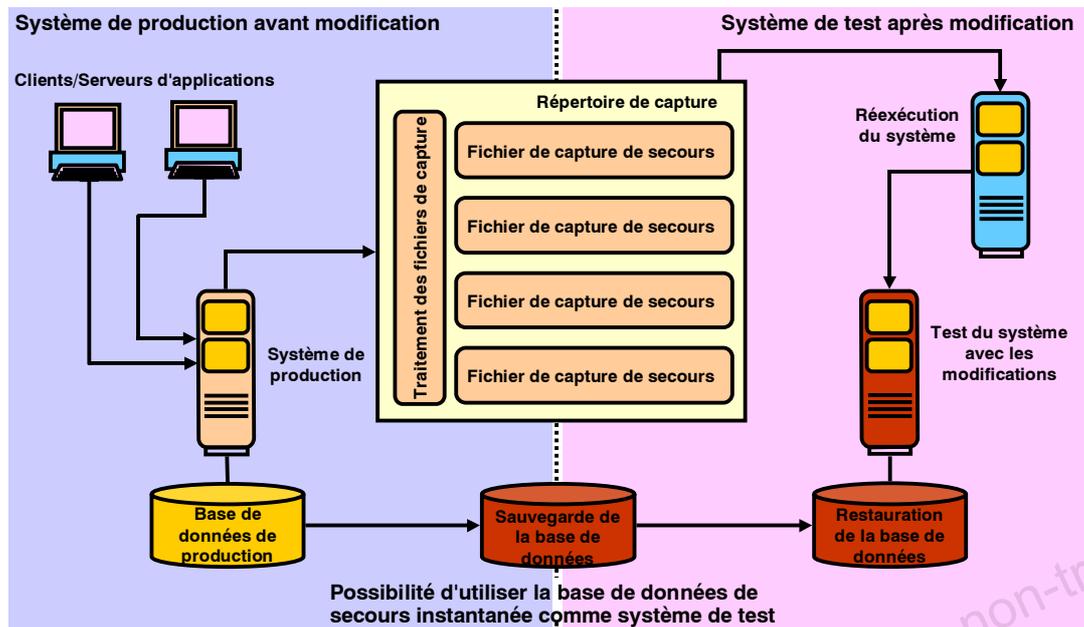
A cet effet, vous créez un environnement de test et testez l'application avec une charge globale. Vous apportez ensuite des modifications et effectuez de nouveaux tests jusqu'à ce que vous soyez certain que l'application fonctionnera dans le nouvel environnement.

La totalité de la procédure repose sur le principe que la charge globale de test est représentative de la charge globale de production réelle. En effet, il suffit qu'il manque une seule instruction dans la charge globale de test pour que la mise à niveau risque de se solder par un échec en production.

Database Replay capture la charge globale de production réelle et vous permet de la réexécuter (replay) dans un environnement de test. La réexécution peut être modifiée pour permettre un débit plus élevé, une application asynchrone ou une réexécution plus rapide ou plus lente. La fonctionnalité Database Replay peut également être utilisée pour un réglage fin par exécution répétée du même jeu d'instructions. Vous pouvez ainsi apporter des modifications à l'environnement, puis observer les différences. Cette fonctionnalité permet également la capture et la réexécution de scénarios de test pour la résolution des erreurs.

L'utilisation de charges globales de production réelles permet de s'assurer que la procédure de test est complète et exacte.

Vue d'ensemble



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Vue d'ensemble

Le fait qu'Oracle Database 11g gère les modifications du système garantit le succès de ces dernières. La fonctionnalité d'enregistrement et de réexécution facilite la mise à niveau d'un serveur de base de données. Ainsi, Database Replay peut s'avérer utile pour tester les performances d'une nouvelle configuration de serveur. Supposons que vous utilisiez une base de données monoinstance et souhaitez passer à une configuration RAC (Real Application Clusters). Vous pouvez enregistrer la charge globale sur une période caractéristique, puis configurer un système de test RAC pour réexécution. Pendant la réexécution, vous pouvez identifier les avantages en termes de performances de la nouvelle configuration en effectuant une comparaison avec celles du système enregistré.

En outre, la fonctionnalité Database Replay peut être utilisée pour le débogage.

Vous pouvez enregistrer et réexécuter des sessions émulant un environnement afin de rendre les bugs plus facilement reproductibles. Un autre avantage est la possibilité de tester la facilité de gestion. En effet, les systèmes en autogestion et autodépannage doivent implémenter automatiquement les conseils ("modèles de calcul autonome"). Les itérations de réexécution multiples permettent de tester et de régler l'efficacité ainsi que la stabilité des stratégies de contrôle.

Pour en savoir plus sur cette technologie, suivez le cours *Oracle Database 11g : Tuning*. Vous pouvez aussi consulter les manuels *Oracle Database Performance Tuning Guide* et *Oracle Database PL/SQL Packages and Types Reference*.

Quiz

Sélectionnez les affirmations qui sont vraies concernant la collecte de statistiques :

1. Pour collecter manuellement les statistiques, vous utilisez le package `DBMS_STATS`.
2. Pour collecter les statistiques automatiquement, vous activez les tâches de maintenance automatisées.
3. Vous pouvez importer des statistiques à partir d'une autre base de données.
4. Vous pouvez collecter des statistiques en configurant les paramètres d'initialisation de la base de données.
5. Vous pouvez collecter des statistiques en modifiant manuellement le dictionnaire de données.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Réponses : 1, 2, 3 et 4

Synthèse

Ce chapitre vous a permis d'apprendre à :

- surveiller les performances des sessions et services
- décrire les avantages de la fonctionnalité Database Replay

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Présentation de l'exercice 14 : Surveiller les performances d'une instance

Cet exercice porte sur les points suivants :

- Surveiller les services et les sessions qui consomment le plus de ressources

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Unauthorized reproduction or distribution prohibited. Copyright© 2014, Oracle and/or its affiliates.

Laurent PERPETUE (laurent.perpetue@srr.fr) has a non-transferable license to use this Student Guide.

15

Gérer les performances via le réglage des instructions SQL

Laurent PERPETUE (laurent.perpetue@sr.fr) has a non-transferable license to use this Student Guide

Objectifs

A la fin de ce chapitre, vous pourrez :

- utiliser la fonction de conseil SQL Tuning Advisor pour :
 - identifier les instructions SQL qui utilisent le plus de ressources
 - régler les instructions SQL qui utilisent le plus de ressources
- utiliser la fonction de conseil SQL Access Advisor pour régler une charge globale

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Réglage des instructions SQL

Processus de réglage des instructions SQL

- Identifiez les instructions SQL mal réglées.
- Réglez les différentes instructions.
- Réglez l'application de façon globale.



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Réglage des instructions SQL

En général, l'effort de réglage qui porte le plus ses fruits est le réglage des instructions SQL. Des instructions SQL mal réglées utilisent plus de ressources que nécessaire. Or, ce manque d'efficacité empêche toute évolutivité, utilise davantage de ressources système et de base de données, et augmente les temps de réponse. Pour y remédier, il convient d'identifier les instructions SQL en cause et de les régler. Les instructions peuvent être réglées individuellement, mais la solution qui optimise une instruction peut souvent nuire aux performances de plusieurs autres.

Les instructions SQL qui utilisent le plus de ressources sont par définition celles qui ont besoin d'être réglées. Il s'agit des instructions qui présentent le temps d'exécution le plus long, utilisent le plus de CPU, ou effectuent le plus de lectures physiques ou logiques.

Réglez les différentes instructions en vérifiant les statistiques de l'optimiseur, consultez l'outil Explain Plan pour connaître le chemin d'accès le plus efficace, testez des constructions alternatives d'instructions SQL, et testez les nouveaux index, vues matérialisées et partitions possibles.

Testez l'application de façon globale, à l'aide des instructions SQL réglées.
Les performances globales sont-elles meilleures ?

La méthodologie est bonne, mais fastidieuse. Le réglage d'une instruction n'est pas difficile. En revanche, il peut s'avérer très difficile de tester son impact global sur l'application.

Dans Oracle Database 11g, différentes fonctions de conseil SQL permettent d'identifier et de régler les instructions, de façon individuelle ou globale.

Fonctions de conseil SQL

The screenshot shows the Oracle Advisor Central interface. At the top, there are tabs for 'Advisors' and 'Checkers'. Below this, there are links for various advisors: ADDM, Memory Advisors, SQL Advisors, Automatic Undo Management, MTR Advisor, SQL Performance Analyzer, Data Recovery Advisor, Segment Advisor, and Streams Performance Advisor. A 'Search' section allows filtering by advisory type and task name. A table displays the results of a search for 'SQL Performance Analyzer', showing a task named '1161_1_49' with a status of 'COMPLETED' and a duration of '1' second.

Advisory Type	Task Name	Advisor Runs	Status	Description	User	Status	Start Time	Duration (seconds)	Expires In (days)
SQL Performance Analyzer	1161_1_49	Last 31 Days	All	ADDM auto run:	SYS	COMPLETED	Aug 25, 2009 2:00:49 AM	1	3

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Fonctions de conseil SQL

Oracle Database 11g fournit un ensemble de fonctions de conseil relatives aux instructions SQL : SQL Access Advisor, SQL Tuning Advisor, SQL Performance Analyzer et SQL Repair Advisor. Le référentiel AWR identifie et enregistre des statistiques relatives aux dernières instructions SQL consommant beaucoup de ressources.

La fonction de conseil SQL Tuning Advisor analyse, une par une, les instructions SQL. Elle examine les statistiques, profils SQL, index, vues matérialisées et instructions SQL restructurées. Elle peut être exécutée manuellement à tout moment, mais elle est aussi lancée dans chaque fenêtre de maintenance concernant les dernières instructions SQL très consommatrices. Cliquez sur Automatic SQL Tuning Results pour consulter et implémenter les recommandations. Cette exécution automatique peut être configurée pour implémenter automatiquement des profils SQL recommandés pour les instructions à forte consommation de ressources.

La fonction de conseil SQL Access Advisor prend en compte les modifications appliquées à un ensemble d'instructions SQL et recherche une réelle amélioration en matières de performances. Il peut s'agir d'un ensemble hypothétique d'instructions SQL, d'un ensemble historique ou d'un ensemble créé manuellement.

SQL Performance Analyzer (SPA) peut être utilisé pour prévoir et prévenir tout problème éventuel de performances lorsque des modifications de l'environnement de la base de données affectent la structure des plans d'exécution SQL.

Fonctions de conseil SQL (suite)

La fonction de conseil SQL Repair Advisor est, quant à elle, exécutée à partir de Support Workbench lorsqu'une instruction SQL échoue avec une erreur critique. Une erreur critique produit également un incident. La fonction de conseil SQL Repair Advisor tente de trouver et de recommander un patch SQL. Si aucun patch n'est trouvé, vous pouvez continuer la procédure dans Support Workbench en créant un package d'incident et en le soumettant au support technique Oracle sous la forme d'une "Service Request" (SR).

Résultats du réglage automatique des instructions SQL

Automatic SQL Tuning Result Summary

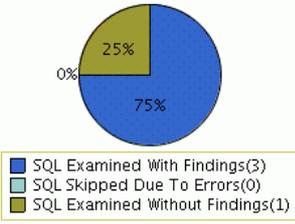
The Automatic SQL Tuning runs during system maintenance windows as an automated maintenance task, searching for ways to improve the execution plans of high-load SQL statements.

Task Status
 Automatic SQL Tuning (SYS_AUTO_SQL_TUNING_TASK) is currently **Enabled** [Configure](#)
 Automatic Implementation of SQL Profiles is currently **Disabled**
 Highly Recommended SQL Profiles **0**

Task Activity Summary
 The activity summary graph shows the benefit of the task activities on the systems high-load SQL. Only profiles that significantly improve SQL performance were implemented.
 Time Period:
 Begin Date: **Jul 4, 2007 10:00:08 PM (UTC+07:00)** End Date: **Jul 6, 2007 2:39:10 AM (UTC+07:00)**

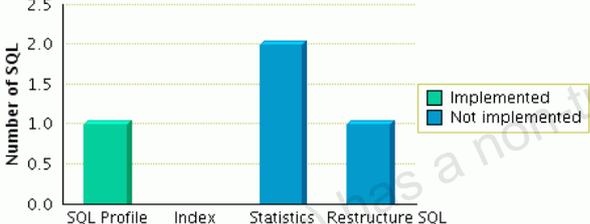
Overall Task Statistics
 Executions **2** Candidate SQL **4** Distinct SQL Examined **4**

SQL Examined Status



Category	Count	Percentage
SQL Examined With Findings	3	75%
SQL Skipped Due To Errors	0	0%
SQL Examined Without Findings	1	25%

Breakdown by Finding Type



Finding Type	Implemented	Not implemented
SQL Profile	1	0
Index	0	2.0
Statistics	0	1.0
Restructure SQL	0	1.0

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Résultats du réglage automatique des instructions SQL

La tâche de réglage automatique des instructions SQL s'exécute par défaut toutes les nuits. Le lien Automatic SQL Tuning Results permet d'accéder au récapitulatif des résultats. En cliquant sur **View Report**, vous pouvez consulter chacune des instructions SQL examinées.

Cliquez sur le bouton Configure pour accéder à une page permettant de modifier les valeurs par défaut de la tâche de réglage automatique des instructions SQL et d'activer l'implémentation automatique des profils SQL.

Oracle Database 11g : Administration Workshop II 15 - 6

Implémenter les recommandations de réglage automatique

Automatic SQL Tuning Result Details

Begin Date Jul 4, 2007 10:00:08 PM (UTC+07:00) End Date Jul 6, 2007 2:39:10 AM (UTC+07:00)

Recommendations

Only profiles that significantly improve SQL performance were implemented.

[View Recommendations...](#) [Implement All](#)

Select	SQL Text	Parsing Schema	SQL ID	Statistics	SQL Profile	Index	Restructure SQL	Miscellaneous	Error	Date
<input checked="" type="radio"/>	SELECT NULL AS table_cat, t.owner...	SYSMAN	Qprhvny3f97z	✓	(82.9%) ✓					7/5/07

Recommendations for SQL ID:Qprhvny3f97z [Return](#)

Only one recommendation should be implemented.

SQL Text

[SELECT NULL AS table_cat, t.owner AS table_schema, t.table_name AS table_name, t.column_name AS column_name, DECODE\(t.data_type, 'CHAR', 1, 'VARCHAR2', 12, 'NUMBER', 3, ...](#)

Select Recommendation

[Original Explain Plan \(Annotated\)](#)

[Implement](#)

Select	Type	Findings	Recommendations	Rationale	Benefit (%)	New Explain Plan	Compare Explain Plans
<input checked="" type="radio"/>	SQL Profile	A potentially better execution plan was found for this statement.	Consider accepting the recommended SQL profile.		82.87	⌕	⌕

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Implémenter les recommandations de réglage automatique

En cliquant sur View Report dans la page Automatic Tuning Results Summary, vous pouvez consulter le détail des résultats du réglage automatique des instructions SQL. Vous pouvez implémenter l'ensemble des recommandations ou procéder à une hiérarchisation descendante afin de consulter ou d'implémenter des recommandations particulières. Dans la page Recommendations, vous pouvez cliquer sur l'icône à droite représentant des lunettes afin de voir les modifications qu'apporterait l'implémentation d'un profil SQL dans l'outil Explain Plan.

SQL Tuning Advisor : Présentation

Réglage des instructions SQL en mode Comprehensive



SQL Tuning
Advisor

Détecter les
statistiques obsolètes
ou manquantes

Régler le plan des
instructions SQL
(profil SQL)

Ajouter les index
manquants

Restructurer les
instructions SQL

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

SQL Tuning Advisor : Présentation

SQL Tuning Advisor est l'élément principal du processus de réglage (tuning). Il effectue plusieurs types d'analyse :

- **Analyse des statistiques** : Il vérifie chaque objet d'interrogation pour déterminer si certaines statistiques sont obsolètes ou manquantes, et il émet des recommandations pour la collecte des statistiques appropriées.
- **Profilage (profiling) des instructions SQL** : L'optimiseur vérifie ses propres estimations et collecte des informations auxiliaires pour éliminer les erreurs possibles. Il construit un profil SQL à l'aide de ces informations auxiliaires et émet une recommandation pour sa création. La création d'un profil SQL permet à l'optimiseur d'instructions de générer un plan bien réglé.
- **Analyse des chemins d'accès** : La création d'index est envisagée si elle améliore l'accès aux tables interrogées dans une mesure importante. Si tel est le cas, des recommandations concernant la création d'objets de ce type sont émises.
- **Analyse de la structure des instructions SQL** : Les instructions SQL qui utilisaient des plans inappropriés sont identifiées et des suggestions concernant leur restructuration sont émises. Les modifications suggérées peuvent être d'ordre syntaxique ou sémantique.

La fonction SQL Tuning Advisor considère indépendamment chaque instruction SQL incluse dans la tâche de conseil. Or, la création d'un nouvel index peut, certes, optimiser une interrogation, mais également avoir un impact négatif sur le temps de réponse des instructions LMD. Un index ou tout autre objet recommandé doit donc être examiné avec la fonction SQL Access Advisor par rapport à une charge globale (ensemble d'instructions SQL) afin de déterminer s'il permet une véritable amélioration des performances.

Utiliser SQL Tuning Advisor

- Utilisez SQL Tuning Advisor pour analyser les instructions SQL et obtenir des recommandations concernant les performances.
- Sources d'analyse pour SQL Tuning Advisor :
 - Top Activity : analyse les principales instructions SQL actuellement actives
 - SQL Tuning Sets : analyse d'un ensemble d'instructions SQL que vous fournissez
 - Historical SQL (AWR) : analyse les instructions SQL à partir des instructions collectées par les clichés AWR

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Utiliser SQL Tuning Advisor

La fonction de conseil SQL Tuning Advisor s'exécute automatiquement toutes les nuits comme la tâche de réglage automatique des instructions SQL. Cependant, une instruction SQL peut parfois nécessiter une action de réglage immédiate. Vous pouvez utiliser la fonction de conseil STA (SQL Tuning Advisor) pour analyser les instructions SQL et obtenir, à tout moment, des recommandations concernant les performances. En général, vous exécutez cette fonction pour évaluer les performances ADDM.

Vous pouvez également exécuter SQL Tuning Advisor lorsque vous souhaitez analyser les instructions SQL qui consomment le plus de temps CPU, d'E/S et de mémoire.

Même si vous pouvez soumettre plusieurs instructions dans une tâche unique, chacune d'elles est analysée indépendamment. Pour obtenir des recommandations de réglage qui tiennent compte des performances globales d'un jeu d'instructions SQL, utilisez la fonction de conseil SQL Access Advisor.

Options de la fonction de conseil SQL Tuning Advisor

Schedule SQL Tuning Advisor [Cancel] [Submit]

Specify the following parameters to schedule a job to run the SQL Tuning Advisor.

* Name:

Description:

* SQL Tuning Set:

SQL Tuning Set Description:

SQL Statements Counts:

Scope

Total Time Limit (minutes):

Scope of Analysis:

Limited
The analysis is done without SQL Profile recommendation and takes about 1 second per statement.

Comprehensive
This analysis includes SQL Profile recommendation, but may take a long time.

Time Limit per Statement (minutes):

Overview

The SQL Tuning Advisor analyzes individual SQL statements, and suggests indexes, SQL profiles, restructured SQL, and statistics that improve the performance of the SQL statements.

The SQL Tuning Advisor operates on a collection of SQL. You can choose a SQL Tuning Set to run the advisor. If you do not have a SQL Tuning Set with the desired SQL for running the advisor, you can create a new one.

You can click on one of the following sources, which will lead you to a data source where you can tune SQL statements using the SQL Tuning Advisor.

[Top Activity](#) [Historical SQL \(AWR\)](#) [SQL Tuning Sets](#)

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Options de la fonction de conseil SQL Tuning Advisor

Dans la page Schedule SQL Tuning Advisor, vous pouvez choisir les instructions SQL à inclure et modifier les valeurs par défaut d'une tâche de réglage. Vous pouvez définir la source des instructions SQL et, si vous disposez du privilège système ADVISOR, vous pouvez soumettre la tâche. Enterprise Manager crée ensuite une tâche de réglage pour la fonction de conseil SQL Tuning Advisor.

Vous pouvez choisir une ou plusieurs instructions SQL récentes sous Top Activity, sélectionner Historical SQL (AWR) ou sélectionner un SQL Tuning Set (STS – Ensemble de réglages SQL) que vous avez déjà créé.

Il est important de choisir la portée appropriée pour la tâche de réglage. Si vous choisissez l'option Limited, SQL Tuning Advisor émet seulement des recommandations basées sur l'analyse des statistiques, l'analyse des chemins d'accès et l'analyse de la structure des instructions SQL. Aucune recommandation concernant le profil des instructions SQL n'est générée. Si vous choisissez l'option Comprehensive, SQL Tuning Advisor émet les mêmes recommandations qu'avec l'option Limited, mais il appelle l'optimiseur en mode de profilage (profiling) des instructions SQL afin de construire un profil SQL. Avec cette option, vous pouvez indiquer une durée maximale pour la tâche de réglage, la valeur par défaut étant de 30 minutes. Après avoir sélectionné Run SQL Tuning Advisor, configurez la tâche de réglage dans la page SQL Tuning Options.

Recommandations SQL Tuning Advisor

SQL Tuning Results:SQL_TUNING_1183667475959

Page Refreshed Jul 6, 2007 3:31:31 AM GMT+07:00

Status **COMPLETED**
 Started Jul 6, 2007 3:31:23 AM
 Completed Jul 6, 2007 3:31:28 AM

Tuning Set Owner **SYS**
 Tuning Set Name **TOP_SQL_1183667475554**
 Time Limit (seconds) **1800**
 Running Time (seconds) **5**

Recommendations

Select	SQL Text	Parsing Schema	SQL ID	Statistics	SQL Profile	Index	Restructure SQL	Miscellaneous	Error
<input checked="" type="radio"/>	DECLARE table_nonexistent EXCEPTION; PRAGMA EXCEPTION_INIT (table_nonexistent, -942); BEGIN BEG...	HR	6t4uxuxdaxpff					✓	
<input type="radio"/>	delete from sh.sales_copy	SYSTEM	0qqwcxx1quwuv	✓					
<input type="radio"/>	insert into sh.sales_copy select * from sh.sales	SYSTEM	axn4pkyvbt51a	✓					

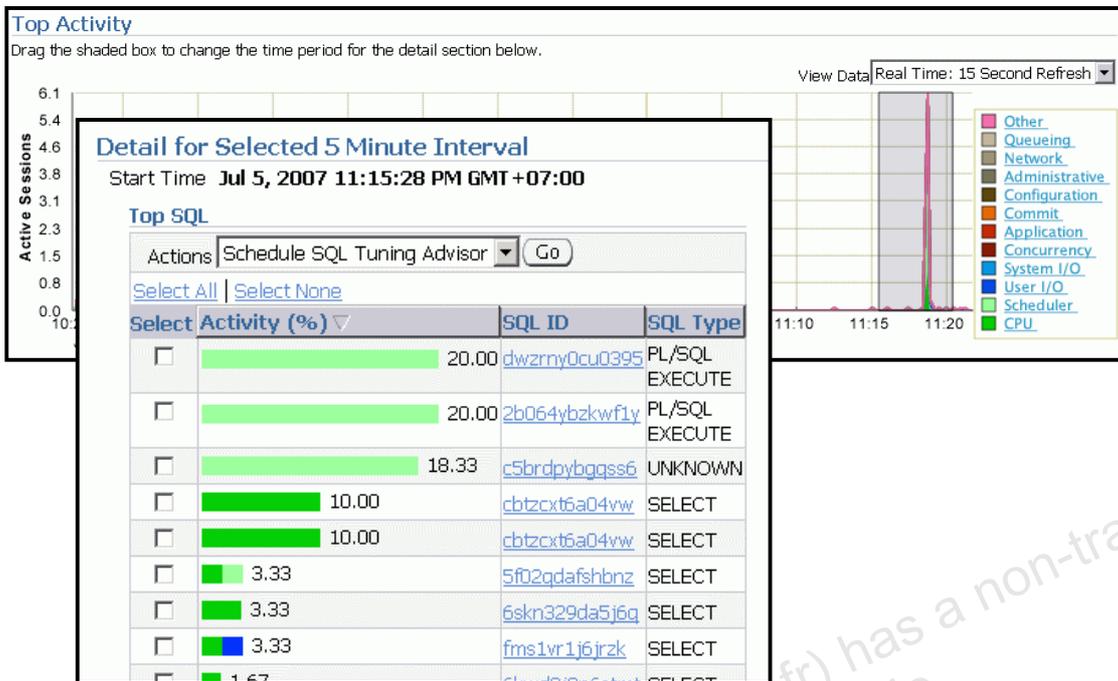
ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Recommandations SQL Tuning Advisor

Les résultats du réglage des instructions SQL sont affichés dès la fin de l'exécution de la tâche. En outre, ils sont accessibles ultérieurement à partir de la page Advisor Central. Un récapitulatif des recommandations est présenté. Vous pouvez consulter et implémenter des recommandations particulières. Pour ce faire, sélectionnez une instruction et cliquez sur View.

Utiliser SQL Tuning Advisor : Exemple



Copyright © 2009, Oracle. Tous droits réservés.

ORACLE

Utiliser SQL Tuning Advisor : Exemple

Vous pouvez appeler SQL Tuning Advisor en procédant comme suit :

1. Cliquez sur Advisor Central dans la région Related Links de la page d'accueil de la base de données.
2. Cliquez sur SQL Advisors. La page SQL Tuning Advisor Links apparaît.
Vous avez le choix entre les options suivantes pour indiquer la source des instructions à analyser :
 - **Active SQL** : Analyse des instructions SQL les plus consommatrices de ressources actuellement actives.
 - **SQL Tuning Sets** : Analyse d'un ensemble d'instructions SQL que vous fournissez.
 - **Historical SQL (AWR)** : Analyse des instructions SQL capturées par des clichés (snapshots) dans le référentiel AWR.
3. Sélectionnez Active SQL. Sélectionnez un intervalle d'analyse de cinq minutes en le désignant à l'aide de la zone ombrée. Sélectionnez les instructions à analyser au cours de la période choisie.
4. Cliquez sur Run SQL Tuning Advisor. Vous accédez à la page SQL Tuning Options, qui affiche les instructions SQL dans l'intervalle. Attribuez un nom et une description à la tâche, sélectionnez la portée Comprehensive et l'heure de début Immediately. Cliquez sur OK.
5. Revenez à la page Advisor Central. Le statut des tâches de conseil est indiqué sous l'en-tête Advisor Tasks dans la région Results. Attendez que le statut de la tâche soit Completed. Vérifiez le statut en cliquant sur le bouton Actualiser de votre navigateur. Sélectionnez la tâche et cliquez sur View Result. La page SQL Tuning Result apparaît.
6. Sélectionnez l'instruction SQL et cliquez sur View Recommendations.

Instructions SQL en double

Duplicate SQL

Page Refreshed Jul 6, 2007 4:49:59 AM GMT+07:00 [Refresh](#)

Applications can cause database to consume excessive CPU by parsing SQL statements that are similar and that can share the same SQL text. Such applications can also cause slow performance by creating contention in the database for Library Cache or Shared Pool.

CPU Consumption Since Instance Started

CPU Used as percentage of Total CPU (%) **2.19**
CPU Used for Parsing as percentage of CPU Used(%) **17.20**

Duplicate SQL Statements

This report identifies similar SQL statements that could be shared by a single SQL statement if the database application used bind variables to replace literals and SQL coding conventions to remove differences based only on character case or whitespace. You can re-write the SQL statements to gain the efficiency of using a single, shared statement.

Note: Only the first 2000 SQL statements that are executed only once are examined. The actual number of SQL statements that are duplicates can be more than 2000.

[Expand All](#) | [Collapse All](#)

Duplicates	Plan Hash Value	SQL Text
▼ Duplicates		
▼ 5	1445457117	select * from hr.employees where employee_id = 148
	1445457117	select * from hr.employees where employee_id = 148
	1445457117	select * from hr.employees where employee_id = 145
	1445457117	select * from hr.employees where employee_id = 133
	1445457117	select * from hr.employees where employee_id = 100
	1445457117	select * from hr.employees where employee_id = 132

Variables attachées candidates

ORACLE

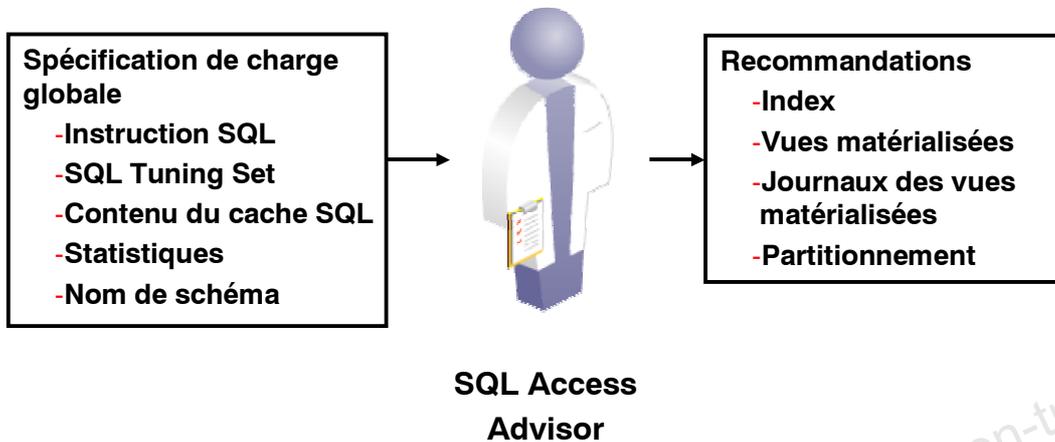
Copyright © 2009, Oracle. Tous droits réservés.

Instructions SQL en double

Les instructions SQL en double sont des instructions qui diffèrent uniquement par les valeurs littérales ou le formatage qu'elles utilisent. Les instructions qui sont différentes obtiennent chacune un curseur distinct dans le cache "library". En revanche, les instructions en double peuvent utiliser le même curseur si les valeurs littérales sont remplacées par des variables attachées (bind variables) et que leur formatage est identique.

Vous pouvez identifier les instructions SQL en double en cliquant sur Duplicate SQL dans l'onglet Performance de la section Additional Monitoring Links. Les instructions SQL similaires sont regroupées, sauf si elles présentent des différences de mise en forme ou de littéraux. Vous pouvez ainsi identifier les instructions SQL de l'application qui peuvent être consolidées, de manière à réduire les exigences portant sur le cache "library" et à accélérer l'exécution des instructions.

SQL Access Advisor : Présentation



Copyright © 2009, Oracle. Tous droits réservés.

ORACLE

SQL Access Advisor : Présentation

La fonction de conseil SAA (SQL Access Advisor) peut recommander un ensemble approprié de vues matérialisées, de journaux des vues matérialisées, de partitions et d'index pour une charge globale donnée. Il est essentiel de comprendre le fonctionnement de ces structures et de les utiliser lors de l'optimisation des instructions SQL car elles peuvent améliorer les performances de l'extraction de données de façon significative.

La fonction de conseil SAA recommande les index bitmap, les index basés sur une fonction et les index B-Tree. Les index bitmap offrent un temps de réponse rapide pour de nombreux types d'interrogation ad hoc. Par ailleurs, ils nécessitent moins d'espace de stockage que les autres techniques d'indexation. Les index B-Tree sont les plus couramment utilisés dans les data warehouses pour l'indexation des clés uniques ou quasi-unicques.

Un autre composant de la fonction SAA fournit aussi des recommandations sur la façon d'optimiser les vues matérialisées pour qu'elles puissent faire l'objet d'une régénération de type Fast et tirer parti de la fonctionnalité General Query Rewrite.

Remarque : Pour plus d'informations sur les vues matérialisées et la réécriture d'interrogation, reportez-vous au manuel *Oracle Database Performance Tuning Guide*.

Session SQL Access Advisor typique

ORACLE Enterprise Manager 11g Database Control

Setup Preferences Help Logout Database

Advisor Central > Logged in As SYS

SQL Access Advisor: Initial Options

Select a set of initial options. Cancel Continue

- Verify use of access structures (indexes, materialized views, partitioning, etc) only
- Recommend new access structures
 - Inherit Options from a previously saved Task or Template

Overview
The SQL Access Advisor evaluates SQL statements in a workload Source, and can suggest indexes, partitioning, materialized views and materialized view logs that will improve performance of the workload as a whole.

TIP You are selecting the starting point for the wizard. All options can be changed from within the wizard.

Copyright © 2009, Oracle. Tous droits réservés.

ORACLE

Session SQL Access Advisor typique

Lorsque vous ouvrez une session de la fonction de conseil SAA (SQL Access Advisor), vous pouvez sélectionner Use Default Options et commencer avec le jeu prédéfini d'options de conseil recommandées. En outre, vous pouvez lancer une tâche et la faire hériter d'un jeu d'options défini par un modèle ou par une tâche, en sélectionnant "Inherit Options from a Task or Template". Plusieurs modèles génériques sont proposés, pour les bases de données généralistes, pour les environnements OLTP et pour les data warehouses. Vous pouvez aussi enregistrer des modèles personnalisés utilisés lors d'une tâche précédente et les réutiliser lorsque vous le souhaitez.

Cliquez sur Continue pour lancer l'assistant de la fonction de conseil SAA.

Remarque : Vous pouvez accéder à la fonction de conseil SAA à partir de la page Advisor Central de Database Control.

Source de la charge globale

SQL Access Advisor: Workload Source Cancel Step 1 of 4 **Next**

Database **orcl**
Logged In As **sys**

Select the source of the workload that you want to use for the analysis. The best workload is one that fully represents all the SQL statements that access the underlying tables.

- Current and Recent SQL Activity**
SQL will be selected from the cache.
- Use an existing SQL Tuning Set**
SQL Tuning Set
- Create a Hypothetical Workload from the Following Schemas and Tables**
The advisor can create a hypothetical workload if the tables contain dimension or primary/foreign key constraints.
Schemas and Tables Add

Comma-separated list

TIP Enter a schema name to specify all the tables belonging to that schema.

[Filter Options](#)

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Source de la charge globale

Utilisez la page Workload Source de l'assistant de la fonction de conseil SAA (SQL Access Advisor) pour désigner la charge globale à utiliser pour la génération des recommandations. Les sources de charge globale prises en charge sont les suivantes :

- **Current and Recent SQL Activity** : utilise les instructions SQL figurant actuellement dans la mémoire cache en tant que charge globale.
- **Use an existing SQL Tuning Set** : vous permet de désigner un SQL Tuning Set (STS - Ensemble de réglages SQL) précédemment créé en tant que source de la charge globale.
- **Create a Hypothetical Workload from the Following Schemas and Tables** : fournit un schéma qui permet à la fonction de conseil de rechercher des tables de dimension et de produire une charge globale.

La charge globale peut être réduite par l'application d'options de filtrage, accessibles dans la section Filter Options. Ces options vous permettent de réduire la portée des instructions SQL qui sont présentes dans la charge globale. Les filtres sont appliqués à la charge globale par la fonction de conseil pour cibler le réglage (tuning). Vous pouvez effectuer un filtrage sur :

- les instructions SQL les plus consommatrices de ressources,
- des utilisateurs, un identificateur de module ou des actions,
- des tables.

Options relatives aux recommandations

SQL Access Advisor: Recommendation Options

Database **orcl** Cancel Back Step 2 of 4 Next

Logged In As **SYS**

Access Structures to Recommend

- Indexes
- Materialized Views
- Partitioning

Scope

The advisor can run in one of two modes, Limited or Comprehensive. Limited Mode is meant to return quickly after processing the statements with the highest cost, potentially ignoring statements with a cost below a certain threshold. Comprehensive Mode will perform an exhaustive analysis.

- Limited
Analysis will focus on highest cost statements
- Comprehensive
Analysis will be exhaustive

[▶ Advanced Options](#)

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Options relatives aux recommandations

Utilisez la page Recommendation Options pour indiquer si les recommandations de la fonction de conseil (advisor) doivent être limitées à une seule méthode d'accès.

Sélectionnez Indexes, Materialized Views, Partitioning ou une combinaison de ces options dans la section Access Structures to Recommend. Vous pouvez sélectionner Evaluation Only pour évaluer uniquement les structures d'accès existantes. Dans ce mode, la fonction de conseil ne génère pas de nouvelles recommandations, mais fournit des commentaires sur l'utilisation des structures existantes. Cela est utile pour effectuer le suivi de l'efficacité d'un index, d'une vue matérialisée ou d'un journal des vues matérialisées au fil du temps.

Vous pouvez utiliser la section Advisor Mode pour choisir l'un des deux modes d'exécution de la fonction de conseil. Ces modes influent sur la qualité des recommandations fournies ainsi que sur la durée de traitement. En mode complet (Comprehensive Mode), la fonction de conseil analyse un grand nombre de candidats et les recommandations générées sont optimales. En mode partiel (Limited Mode), la fonction de conseil s'exécute rapidement, ce qui limite le nombre de recommandations candidates.

Options relatives aux recommandations

The screenshot shows the 'Advanced Options' configuration page. It is divided into several sections:

- Workload Categorization:** Includes 'Workload Volatility' with a checkbox 'Consider only queries' (with a note: 'Choose this option if this is a Data Warehouse workload') and 'Workload Scope' with a checkbox 'Recommend dropping unused access structures' (with a note: 'Select when workload represents all access structure use cases').
- Space Restrictions:** Asks 'Do you want to limit additional space used by recommended indexes and materialized views?'. It has two radio buttons: 'No, show me all recommendations (unlimited space)' (selected) and 'Yes, limit additional space to' followed by a text input field and a 'MB' dropdown menu. A note below says: 'Set to zero or negative to recommend dropping existing access structures to make room for better ones.'
- Tuning Prioritization:** 'Prioritize tuning of SQL statements by' has a dropdown menu with 'PRIORITY,OPTIMIZER_COST' selected, an 'Add' button, and a 'Buffer Gets' dropdown menu. Below it, a note says: 'SQL statements will be analyzed in descending order of the value of the prioritized statistic.' There is a checked checkbox 'Consider access structures creation costs recommendations' with a note: 'If checked, the SQL Access Advisor will weigh the cost of creation of access structures against the frequency and potential improvement of SQL statement execution time. If unchecked, the cost of creation will be ignored. Check this box if you do not want specific recommendations generated for statements that are not executed frequently.'
- Default Storage Locations:** Explains that by default, indexes and materialized views are placed in the schema and tablespace of the table they reference. It lists several fields for configuration: 'Index Tablespace', 'Index Schema', 'Materialized View Tablespace', 'Materialized View Schema', 'Materialized View Log Tablespace', and 'Partitioning Tablespaces'. Each field has a text input box and a blue pencil icon to its right. A note at the bottom says 'Comma-separated list'.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Options relatives aux recommandations (suite)

Vous pouvez choisir Advanced Options pour afficher ou masquer les options qui vous permettent de définir les restrictions d'espace, les options de réglage (tuning) et les emplacements de stockage par défaut. Utilisez la section Workload Categorization pour définir les options Workload Volatility et Workload Scope. Vous pouvez décider de favoriser les opérations en lecture seule ou tenir compte de la volatilité des objets référencés lorsque vous générez des recommandations. Vous pouvez également sélectionner Partial Workload, afin d'exclure les recommandations consistant à supprimer les structures d'accès inutilisées, ou Complete Workload, pour inclure ces recommandations.

Dans la section Space Restrictions, définissez une limite d'espace sur le disque dur. La fonction de conseil génère alors ses recommandations en veillant à ce que leur implémentation ne dépasse pas cette limite.

Dans la section Tuning Options, vous pouvez personnaliser les recommandations générées par la fonction de conseil. Utilisez la liste déroulante "Prioritize Tuning of SQL Statements by" pour classer les instructions selon l'un des paramètres suivants : Optimizer Cost, Buffer Gets, CPU Time, Disk Reads, Elapsed Time ou Execution Count.

Dans la section Default Storage Locations, vous pouvez remplacer les emplacements par défaut définis pour le schéma et pour le tablespace. Par défaut, les index sont placés dans le schéma et dans le tablespace de la table à laquelle ils font référence. Les vues matérialisées sont placées dans le schéma et dans le tablespace de l'utilisateur qui a exécuté l'une des interrogations ayant contribué à la recommandation relative à la vue matérialisée.

Une fois que vous avez défini ces paramètres, vous pouvez planifier et vérifier la tâche de réglage.

Examiner les recommandations

Results for Task: SQLACCESS1524225

Task Name: SQLACCESS1524225
 Status: COMPLETED
 Advisor Mode: COMPREHENSIVE
 Scheduler Job: ADV_SQLACCESS1524225
 Publish Point: 1
 Running Time: [blank]
 Total Time Limit: [blank]

Overall Workload Performance

Potential for Improvement

Workload I/O Cost

Workload I/O Cost

Original Cost (35777384)
 New Cost (35766843)

Query Execution

% of Statements vs Query Improvement Factor

Recommendations

Recommendations: 1
 Space Requirements (MB): 0.008
 User Specified Space Adjustment: Unlimited

Recommendations by Cost Improvement

Cost Improvement vs ID

Select Recommendations for Implementation

Implementation Status	ID	Actions	Action Types	Cost Improvement	Cost Improvement (%)	Estimated Space Used (MB)	Affected SQL Statements
<input checked="" type="checkbox"/>	1	1		10541	100.00	0.008	1

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Examiner les recommandations

Dans la page Advisor Central, vous pouvez afficher toutes les tâches de réglage de la fonction de conseil SAA (SQL Access Advisor) qui sont terminées. Sélectionnez celle dont vous souhaitez afficher les recommandations, puis cliquez sur le bouton View Result. Utilisez l'onglet Summary de la page Results for Task pour obtenir une vue générale des résultats de la fonction de conseil. Cette page contient des graphiques et des statistiques qui présentent l'amélioration potentielle globale des performances et du temps d'exécution des interrogations pour les recommandations proposées. Vous pouvez utiliser cette page pour connaître le nombre d'instructions et le nombre d'actions recommandées.

Pour consulter d'autres aspects des résultats de la tâche de conseil, cliquez sur l'un des trois autres onglets de la page : Recommendations, SQL Statements ou Details.

La page Recommendations contient un graphique et un tableau qui affichent les principales recommandations, classées en fonction du pourcentage d'amélioration apporté au coût total de la charge globale. Les principales recommandations sont celles qui offrent la plus forte amélioration totale des performances.

En cliquant sur le bouton Show SQL, vous pouvez voir le script SQL généré pour les recommandations sélectionnées. Vous pouvez cliquer sur l'identificateur d'une recommandation dans le tableau afin de voir la liste des actions à effectuer pour son implémentation. Dans la page Actions, vous pouvez voir toutes les instructions SQL à exécuter pour implémenter l'action. Ne cochez pas les cases correspondant aux recommandations que vous ne souhaitez pas implémenter. Cliquez ensuite sur le bouton Schedule Implementation afin d'implémenter les actions sélectionnées. Cette étape est exécutée en tant que travail du planificateur.

SQL Performance Analyzer : Présentation

- Utilisateurs ciblés : DBA, responsables de la qualité, développeurs d'application
- Aide à prévoir l'impact des modifications du système sur le temps de réponse pour la charge SQL globale
- Crée différentes versions des performances de la charge SQL globale (plans et statistiques d'exécution SQL)
- Exécute les instructions SQL en série (pas de traitement simultané)
- Analyse les différences de performances
- Fournit une analyse détaillée des performances de chaque instruction SQL
- Intégré à SQL Tuning Advisor pour le réglage des instructions dont les performances ont régressé

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

SQL Performance Analyzer : Présentation

Oracle Database 11g inclut la fonctionnalité SQL Performance Analyzer qui fournit une évaluation précise de l'impact des modifications sur les instructions SQL composant la charge globale. SQL Performance Analyzer vous aide à prévoir l'impact d'une modification potentielle sur les performances d'une charge SQL. Les DBA peuvent ainsi consulter des informations détaillées sur les performances des instructions SQL, notamment les statistiques avant et après exécution, et la liste des instructions dont les performances se sont améliorées ou dégradées. Cela vous permet de modifier un environnement de test de façon à déterminer si la mise à niveau d'une base améliorera les performances de la charge globale.

SQL Performance Analyzer : Cas d'utilisation

SQL Performance Analyzer est avantageux dans les cas suivants :

- Mise à niveau d'une base de données
- Implémentation de recommandations de réglage
- Modification au niveau schéma
- Collecte de statistiques
- Modification des paramètres d'une base de données
- Modification du système d'exploitation et du matériel

Accessible via Enterprise Manager et le package DBMS_SQLPA



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

SQL Performance Analyzer : Cas d'utilisation

SQL Performance Analyzer peut être utilisé pour prévoir et prévenir tout problème éventuel de performances lorsque des modifications de l'environnement de la base de données affectent la structure des plans d'exécution SQL. Voici quelques exemples de ces modifications :

- Mise à niveau d'une base de données
- Implémentation de recommandations de réglage
- Modification au niveau schéma
- Collecte de statistiques
- Modification des paramètres d'une base de données
- Modification du système d'exploitation et du matériel

Vous pouvez utiliser SQL Performance Analyzer pour prévoir l'impact des modifications sur les performances des instructions SQL, même dans les environnements les plus complexes. Les développeurs d'applications de base de données peuvent y recourir pour tester les modifications apportées aux schémas, aux objets de base de données et aux applications réécrites afin d'évaluer leur impact sur les performances.

SQL Performance Analyzer vous permet également de comparer les statistiques relatives aux performances des instructions SQL.

Vous pouvez accéder à SQL Performance Analyzer via Enterprise Manager ou à l'aide du package DBMS_SQLPA.

Pour plus d'informations, suivez les cours *Oracle Database 11g : Tuning* et *Oracle Database 11g : Nouvelles fonctionnalités d'administration*. Pour plus d'informations sur le package DBMS_SQLPA, consultez le manuel *Oracle Database PL/SQL Packages and Types Reference Guide*.

Utiliser SQL Performance Analyzer

1. Capturez la charge SQL globale dans le système de production.
2. Transportez cette charge dans un système de test.
3. Générez les données de performances "avant modification".
4. Procédez aux modifications.
5. Générez les données de performances "après modification".
6. Comparez les résultats des étapes 3 et 5.
7. Réglez les instructions SQL dont les performances ont régressé.



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Utiliser SQL Performance Analyzer

1. **Collecte SQL** : Dans cette phase, vous collectez l'ensemble des instructions SQL qui composent la charge SQL globale dans le système de production.
2. **Transport** : Vous devez transporter la charge globale obtenue vers le système de test. Le STS est exporté du système de production, puis importé dans le système de test.
3. **Evaluation des performances "avant modification"** : Avant d'apporter des modifications, vous exécutez les instructions SQL pour collecter les informations de référence qui permettront d'évaluer l'impact d'une future modification sur les performances de la charge globale.
4. **Modification** : Une fois en possession des données "avant modification", vous pouvez implémenter la modification prévue et commencer à examiner son impact sur les performances.
5. **Evaluation des performances "après modification"** : Cette étape s'effectue après la modification de l'environnement de base de données. Chaque instruction de la charge SQL globale est exécutée en mode simulation (collecte des statistiques uniquement). Les informations capturées sont les mêmes qu'à l'étape 3.
6. **Comparaison et analyse des performances SQL** : Une fois que vous avez les deux versions des données de performances pour la charge SQL globale, vous pouvez procéder à une analyse par comparaison.
7. **Réglage des instructions SQL dont les performances ont régressé** : A ce stade, vous avez identifié précisément les instructions SQL responsables des problèmes de performances suite à la modification de la base. Vous pouvez utiliser n'importe quel outil de base de données pour effectuer le réglage du système. Après l'implémentation des actions de réglage, répétez la procédure pour créer une nouvelle version "après modification" et vérifiez que les nouvelles performances sont satisfaisantes.

Quiz

SQL Tuning Advisor doit toujours être lancé séparément, même lorsque les tâches de maintenance automatisées sont activées.

1. Vrai
2. Faux

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Réponse : 2

Quiz

Vous pouvez recevoir des recommandations relatives aux performances pour les instructions Historical SQL collectées par les clichés AWR.

1. Vrai
2. Faux

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Réponse : 1

Quiz

La fonction de conseil SAA (SQL Access Advisor) peut recommander un ensemble approprié de vues matérialisées, de journaux des vues matérialisées, de partitionnement et d'index pour une charge globale donnée.

1. Vrai
2. Faux

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Réponse : 1

Quiz

SQL Performance Analyzer fournit des informations détaillées sur les performances des instructions SQL, telles que les statistiques avant et après exécution, et les instructions dont les performances ont progressé ou régressé.

1. Vrai
2. Faux

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Réponse : 1

Synthèse

Ce chapitre vous a permis d'apprendre à :

- utiliser la fonction de conseil SQL Tuning Advisor pour :
 - identifier les instructions SQL qui utilisent le plus de ressources
 - régler les instructions SQL qui utilisent le plus de ressources
- utiliser la fonction de conseil SQL Access Advisor pour régler une charge globale

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Présentation de l'exercice 15 : Régler les instructions SQL pour améliorer les performances

Cet exercice porte sur les points suivants :

- Utiliser SQL Tuning Advisor

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

16

Gérer les ressources

Copyright © 2009, Oracle. Tous droits réservés.

ORACLE

Laurent PERPETUE (laurent.perpetue@sr.fr) has a non-transferable license to use this Student Guide

Objectifs

A la fin de ce chapitre, vous pourrez :

- configurer Database Resource Manager
- créer des plans d'allocation de ressources et y accéder
- créer des groupes de consommateurs de ressources
- définir des directives pour l'allocation de ressources à des groupes de consommateurs
- associer des groupes de consommateurs de ressources à des plans
- activer un plan d'allocation de ressources
- surveiller Resource Manager

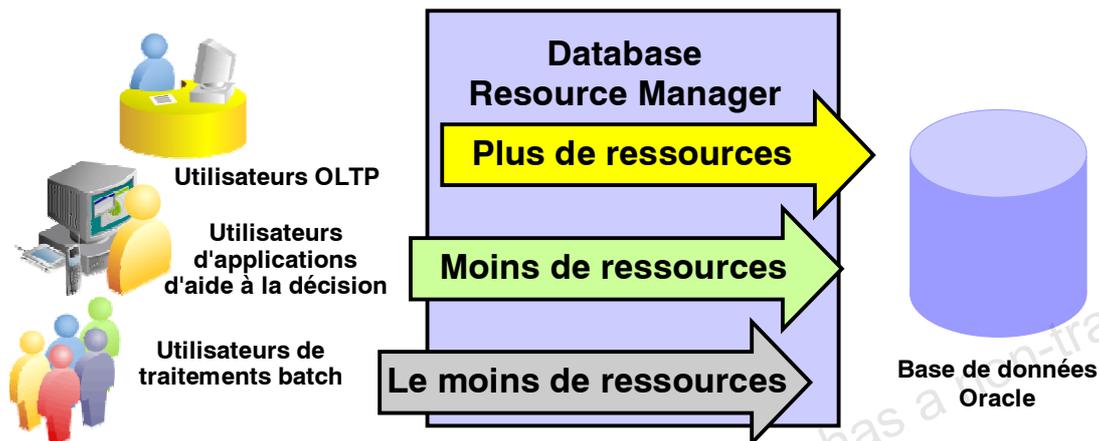
ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Database Resource Manager : Présentation

Utilisez Resource Manager pour :

- gérer des charges hétérogènes
- contrôler les performances du système



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Database Resource Manager : Présentation

Avec Database Resource Manager (également appelé Resource Manager), vous pouvez exercer plus de contrôle sur l'allocation des ressources informatiques que dans le cadre des fonctions de gestion des ressources classiques fournies par le système d'exploitation.

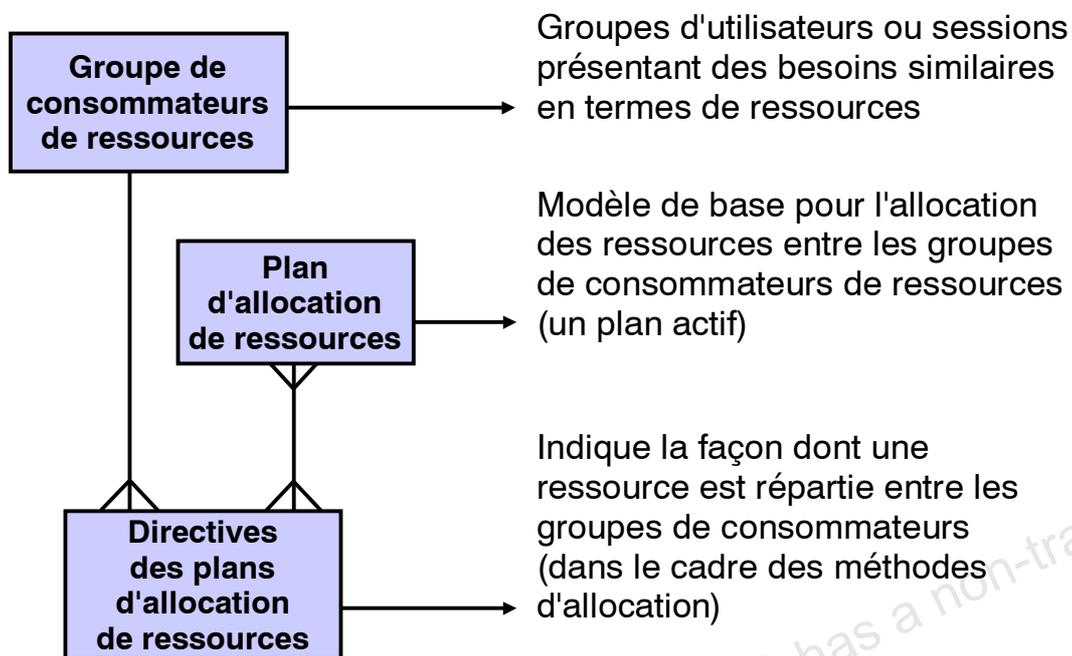
Les décisions prises par ce dernier peuvent conduire à des problèmes de différents types :

- Surcharge excessive due au changement de contexte du système d'exploitation par les processus serveur de la base de données Oracle lorsque le nombre de processus serveur est élevé.
- Suspension d'un processus serveur de la base de données qui détient un verrou interne (latch).
- Distribution inégale des ressources entre tous les processus de base de données Oracle, et impossibilité d'affecter une priorité à une tâche par rapport à une autre.
- Impossibilité de gérer les ressources propres à la base de données, par exemple les serveurs d'exécution en parallèle et les sessions actives.

Database Resource Manager contrôle la distribution des ressources entre les différentes sessions en assurant le suivi de la planification de l'exécution à l'intérieur de la base de données. En contrôlant l'exécution des sessions et leur durée, Database Resource Manager fait en sorte que la distribution des ressources corresponde à la directive du plan et, par conséquent, aux objectifs de l'entreprise. Ainsi, il est possible de garantir aux groupes d'utilisateurs une quantité minimale de ressources de traitement, indépendamment de la charge du système et du nombre d'utilisateurs.

Le package `DBMS_RESOURCE_MANAGER_PRIVS` contient les procédures permettant d'octroyer et de révoquer le privilège système `ADMINISTER_RESOURCE_MANAGER`, qui est un prérequis pour l'appel de Resource Manager.

Database Resource Manager : Concepts



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Database Resource Manager : Concepts

L'administration des systèmes à l'aide de Database Resource Manager implique l'utilisation de plans d'allocation de ressources, de groupes de consommateurs de ressources et de directives de plan d'allocation de ressources.

Un *groupe de consommateurs de ressources* est un ensemble d'utilisateurs ou de sessions qui ont des besoins similaires en termes d'utilisation des ressources système et des ressources de base de données.

Un *plan d'allocation de ressources* indique comment les ressources sont réparties entre les différents groupes de consommateurs. Database Resource Manager permet également de créer des plans à l'intérieur de plans, appelés *sous-plans*.

Les *directives de plan d'allocation de ressources* indiquent la façon dont une ressource particulière est partagée par les groupes de consommateurs ou les sous-plans. Vous associez des groupes de consommateurs de ressources et des sous-plans à un plan d'allocation de ressources particulier via des directives.

Les *méthodes d'allocation des ressources* déterminent la stratégie à employer pour l'allocation d'une ressource donnée. Elles sont utilisées par les plans d'allocation de ressources et les groupes de consommateurs de ressources.

Pourquoi utiliser Resource Manager

- Vous pouvez gérer les ressources de la base de données et du système d'exploitation :
 - Utilisation de la CPU
 - Degré de parallélisme
 - Nombre de sessions actives
 - Génération des informations d'annulation
 - Durée d'exécution des opérations
 - Temps d'inactivité
 - Consolidation au niveau base de données
 - Consolidation au niveau serveur
- Vous pouvez également définir des critères qui, s'ils sont satisfaits, entraînent le basculement automatique des sessions vers un autre groupe de consommateurs de ressources.

Accès via :

- EM
- le package
DBMS_RESOURCE_MANAGER



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Pourquoi utiliser Resource Manager

Database Resource Manager offre plusieurs moyens d'allouer des ressources :

- **Méthode d'allocation de CPU** : permet d'indiquer la façon dont les ressources CPU sont allouées entre les groupes de consommateurs de ressources et les sous-plans.
- **Limite relative au degré de parallélisme** : permet de contrôler le degré maximal de parallélisme de n'importe quelle opération dans un groupe de consommateurs de ressources.
- **Pool de sessions actives avec mise en file d'attente** : permet de limiter le nombre de sessions actives simultanées pour un groupe de consommateurs de ressources ou un sous-plan. Si un groupe dépasse le nombre maximum de sessions autorisées, les nouvelles sessions sont placées dans une file d'attente, où elles attendent la fin d'une session active. Vous pouvez également définir une limite quant à la durée d'attente d'une session avant l'arrêt et le renvoi d'une erreur.
- **Pool d'annulation** : permet de contrôler la quantité totale d'informations d'annulation (undo) pouvant être générée par un groupe de consommateurs de ressources ou un sous-plan. Lorsque la valeur globale de l'espace d'annulation dépasse la quantité définie par le paramètre UNDO_POOL, aucune autre commande INSERT, UPDATE ou DELETE n'est autorisée tant que l'espace d'annulation n'a pas été libéré par une autre session du groupe ou tant que l'espace d'annulation n'a pas été augmenté pour le groupe. Si le quota affecté au groupe est dépassé au cours de l'exécution d'une instruction LMD, l'opération est abandonnée et renvoie un message d'erreur. Les interrogations restent autorisées, même si un groupe de consommateurs de ressources a dépassé son seuil d'annulation.

Pourquoi utiliser Resource Manager (suite)

- **Limite relative à la durée d'exécution** : permet de définir une durée maximale d'exécution autorisée pour une opération. La base de données Oracle utilise les statistiques de l'optimiseur basé sur le coût pour évaluer la durée d'exécution d'une opération. Si cette durée est supérieure à la durée maximale autorisée (`MAX_EST_EXEC_TIME`), l'opération renvoie une erreur et n'est pas démarrée. Si un groupe de consommateurs de ressources comporte plusieurs directives de plan définissant `MAX_EST_EXEC_TIME`, Resource Manager choisit la valeur la plus restrictive parmi toutes les valeurs entrantes.
- **Limite relative à la durée d'inactivité** : permet d'indiquer la durée d'inactivité au terme de laquelle une session est arrêtée (`MAX_IDLE_TIME`). Vous pouvez aussi configurer une durée maximale d'inactivité distincte pour les sessions qui bloquent d'autres sessions (`MAX_IDLE_TIME_BLOCKER`).
- **Changement de groupe de consommateurs de ressources** : Le groupe de consommateurs de ressources initial est le groupe auquel une session appartiendrait si elle venait d'être ouverte. La notion d'appel principal consiste à considérer un bloc PL/SQL entier comme un appel unique ou, de la même façon, à considérer comme des appels distincts des instructions SQL exécutées séparément par le client. Cette fonctionnalité est surtout intéressante pour les applications à trois niveaux (3-tiers), dans lesquelles le serveur du niveau intermédiaire (middle tier) implémente des pools de sessions. Dans ce cas, le niveau intermédiaire tend à faire un appel pour un utilisateur final, puis à utiliser la même session pour l'appel d'un utilisateur final différent. Par conséquent, les limites du travail sont en fait les appels, et les actions d'un utilisateur final précédent ne doivent pas affecter l'utilisateur final suivant. Vous pouvez créer une directive du plan d'allocation des ressources de sorte que Resource Manager renvoie automatiquement l'utilisateur vers le groupe de consommateurs de ressources initial à la fin de l'appel principal.

Remarque : Vous ne pouvez pas indiquer les paramètres `SWITCH_TIME_IN_CALL` et `SWITCH_TIME` dans la même directive. Le paramètre `SWITCH_TIME` est essentiellement destiné aux applications client/serveur, tandis que le paramètre `SWITCH_TIME_IN_CALL` concerne les applications à trois niveaux (3-tiers).

- **Consolidation au niveau base de données** : Resource Manager permet d'optimiser l'allocation des ressources entre différentes sessions de base de données simultanées. Cette opération requiert que les applications soient isolées l'une de l'autre. L'augmentation de la charge globale d'une application ne doit pas avoir d'impact sur les autres applications. En outre, les performances de chaque application doivent être régulières. Les applications éligibles pour la consolidation au niveau base de données sont les tâches de maintenance automatisées, car elles peuvent utiliser jusqu'à 100 % des ressources CPU côté serveur.

Pourquoi utiliser Resource Manager (suite)

- **Consolidation au niveau serveur :** De nombreuses bases de données de test, de développement et de production à petite échelle n'utilisent pas toute la puissance des serveurs sur lesquels elles sont exécutées. Une alternative possible est alors la *consolidation au niveau serveur*. En exécutant plusieurs instances de bases de données sur le serveur, l'utilisation des ressources est mieux optimisée. La méthode qui permet de gérer les allocations de CPU sur un serveur multi-CPU exécutant plusieurs instances de base de données est appelée mise en cage d'instance. Comme elle est simple à configurer et ne requiert ni nouvelle licence de logiciel, ni installation supplémentaire, elle constitue une excellente alternative aux autres outils de consolidation au niveau serveur, tels que les gestionnaires de virtualisation et de charge globale du système d'exploitation.

Vous pouvez accéder aux plans de ressources avec l'interface graphique d'Enterprise Manager ou la ligne de commande du package DBMS_RESOURCE_MANAGER.

Plan Resource Manager de maintenance par défaut

```
SQL> show parameter resource_manager_plan
```

NAME	TYPE	VALUE
resource_manager_plan	string	SCHEDULER [0x2843] : DEFAULT_MAINTENANCE_PLAN

Group/Subplan	Level 1	Level 2
ORA\$AUTOTASK_SUB_PLAN	0	25
ORA\$DIAGNOSTICS	0	5
OTHER_GROUPS	0	70
SYS_GROUP	100	0

Group/Subplan	Level 1	Level 2	Level 3
ORA\$AUTOTASK_HIGH_SUB_PLAN	0	100	0
ORA\$AUTOTASK_MEDIUM_GROUP	0	0	100
ORA\$AUTOTASK_URGENT_GROUP	100	0	0

Group/Subplan	Percentage
ORA\$AUTOTASK_HEALTH_GROUP	25
ORA\$AUTOTASK_SPACE_GROUP	25
ORA\$AUTOTASK_SQL_GROUP	25
ORA\$AUTOTASK_STATS_GROUP	25

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Plan Resource Manager de maintenance par défaut

Les tâches de maintenance automatisées requièrent l'activation de Resource Manager dans les fenêtres de maintenance. Au début d'une fenêtre de maintenance, le plan Resource Manager `DEFAULT_MAINTENANCE_PLAN` est automatiquement défini pour contrôler la quantité de CPU utilisée par les tâches de maintenance automatisées. Pour l'attribution de différentes priorités à chaque tâche possible dans une fenêtre de maintenance, différents groupes de consommateurs de ressources sont affectés au plan `DEFAULT_MAINTENANCE_PLAN`. La hiérarchie entre les groupes et les plans est illustrée dans la diapositive ci-dessus.

Pour les tâches de priorité élevée, les affectations sont les suivantes :

- Optimizer Statistics Gathering est dans le groupe de consommateurs `ORA$AUTOTASK_STATS_GROUP`.
- Segment Advisor est dans le groupe de consommateurs `ORA$AUTOTASK_SPACE_GROUP`.
- Automatic SQL Tuning est dans le groupe de consommateurs `ORA$AUTOTASK_SQL_GROUP`.

Remarque : Au besoin, vous pouvez modifier manuellement le pourcentage de ressources CPU allouées à chaque groupe dans le plan `ORA$AUTOTASK_HIGH_SUB_PLAN`.

Exemple : DEFAULT_PLAN

Groupe de consommateurs de ressources	Méthodes d'allocation		
	MGMT_P1	MGMT_P2	MGMT_P3
SYS_GROUP	100%	0%	0%
OTHER_GROUPS	0%	90%	0%
ORA\$AUTOTASK_SUB_PLAN	0%	5%	0%
ORA\$DIAGNOSTICS	0%	5%	0%

Pour les tâches de maintenance automatisées

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Exemple : DEFAULT_PLAN

Le plan d'allocation de ressources DEFAULT_PLAN fait partie des plans par défaut fournis. Il contient des directives destinées aux groupes de consommateurs fournis suivants :

- **SYS_GROUP** : groupe de consommateurs de ressources initial pour les utilisateurs SYS et SYSTEM.
- **OTHER_GROUPS** : utilisé pour toutes les sessions qui appartiennent à des groupes de consommateurs exclus du plan d'allocation de ressources actif. OTHER_GROUPS doit disposer de directives de plan d'allocation de ressources dans un des plans actifs.
- **ORA\$AUTOTASK_SUB_PLAN** : groupe présentant un niveau de priorité inférieur à ceux de SYS_GROUP et OTHER_GROUPS dans le plan.
- **ORA\$DIAGNOSTICS** : groupe de ce plan d'allocation de ressources doté de la même priorité que ORA\$AUTOTASK_SUB_PLAN. La faible priorité des groupes ORA\$ empêche n'importe quel travail de maintenance automatisé de consommer des quantités excessives de ressources système.

Le groupe de consommateurs de ressources initial d'un utilisateur est le groupe auquel appartient initialement toute session créée par cet utilisateur. Si vous n'avez pas défini de groupe initial pour un utilisateur, le groupe DEFAULT_CONSUMER_GROUP est utilisé par défaut.

Le plan DEFAULT_PLAN et les groupes de consommateurs de ressources associés peuvent être utilisés ou non. SYSTEM_PLAN peut servir de modèle pour la création de nouveaux plans d'allocation de ressources. Il peut être modifié ou supprimé. Utilisez-le de façon appropriée, en fonction de l'environnement.

Workflow possible

Workflow pour les objets Resource Manager obligatoires :

- Créer un nouveau plan d'allocation de ressources
- Créer un groupe de consommateurs de ressources
- Affecter des utilisateurs à des groupes
- Définir les directives du plan d'allocation de ressources
- Activer un plan d'allocation de ressources

```
DBMS_RESOURCE_MANAGER.CREATE  
_CONSUMER_GROUP (  
CONSUMER_GROUP => 'APPUSER',  
MGMT_MTH => 'ROUND-ROBIN',  
COMMENT => '');
```

```
DBMS_RESOURCE_MANAGER_PRIVS.  
GRANT_SWITCH_CONSUMER_GROUP  
(  
grantee_name => 'PM',  
consumer_group => 'APPUSER',  
grant_option => FALSE );
```

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Workflow possible

Pour créer un nouveau plan, vous devez configurer plusieurs objets Resource Manager.

Créer un nouveau plan d'allocation de ressources

Le planificateur peut changer automatiquement de plan Resource Manager lorsqu'il arrive en début ou en fin de période. Si cela n'est pas adapté à la situation, désactivez l'option par défaut "Automatic Plan Switching Enabled".

Créer des groupes de consommateurs de ressources

Utilisez la page Resource Consumer Groups d'EM pour créer ou éditer un groupe de consommateurs et sa description, pour ajouter ou supprimer les utilisateurs associés (membres) et pour définir ou modifier les rôles de base de données correspondants.

Indiquez la méthode à utiliser pour la répartition de la CPU entre les sessions composant le groupe de consommateurs de ressources. La méthode Round Robin (tourniquet) garantit une répartition équitable entre les sessions. Il s'agit de la méthode utilisée par défaut.

Avec la méthode Run to Completion, les sessions présentant la durée d'activité la plus importante sont planifiées avant les autres sessions. Vous obtenez le même résultat en utilisant la procédure `DBMS_RESOURCE_MANAGER.CREATE_CONSUMER_GROUP` avec l'option `MGMT_MTH`.

Workflow possible (suite)

Affecter des utilisateurs à des groupes

- Des utilisateurs doivent être affectés aux groupes de consommateurs de ressources. Le groupe de consommateurs par défaut d'un utilisateur est celui auquel appartient initialement toute session créée par cet utilisateur. Si un utilisateur n'est pas associé à un groupe de consommateurs, le groupe `DEFAULT_CONSUMER_GROUP` est défini par défaut. Vous devez octroyer directement à l'utilisateur ou à `PUBLIC` le privilège de changement de groupe (`switch`) sur un groupe de consommateurs de ressources afin qu'il puisse devenir le groupe par défaut de l'utilisateur. Le privilège de changement de groupe ne peut pas provenir d'un rôle accordé à cet utilisateur.
- Le package `DBMS_RESOURCE_MANAGER_PRIVS` contient la procédure permettant d'affecter des groupes de consommateurs de ressources aux utilisateurs. L'octroi du privilège de changement de groupe à un utilisateur permet à cet utilisateur de changer de groupe de consommateurs de ressources.

Définir les directives du plan d'allocation de ressources

Edit Resource Plan: DEFAULT_PLAN

1
2
3
4
5
6
Show SQL
Revert
Apply

General
Parallelism
Session Pool
Undo Pool
Threshold
Idle Time

A Resource Plan contains directives that specify how resources are allocated to Consumer Groups. For each Consumer Group, a directive specifies the amount of CPU resources are allocated. It also specifies limits, such as the maximum degree of parallelism, execution time, and amount of I/O, that each session in the Consumer Group can consume. You can enable a Resource Plan manually or automatically, using Scheduler Windows.

Plan **DEFAULT_PLAN**
 Description

Activate this plan
 Automatic Plan Switching Enabled

Resource Allocations

Mode: Percentage Advanced Modify

Group/Subplan	Level 1	Level 2	Level 3	Level 4	Level 5	Level 6	Level 7	Level 8
APPUSER			60					
LOW_GROUP			40					
ORA\$AUTOTASK_SUB_PLAN		5						
ORA\$DIAGNOSTICS		5						
OTHER_GROUPS		90						
SYS_GROUP	100							

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Définir les directives du plan d'allocation de ressources

Si vous n'utilisez pas Enterprise Manager pour créer le plan d'allocation de ressources ou les groupes de consommateurs de ressources, vous devez commencer par créer une *zone d'attente*. Il s'agit d'une zone vide qui permet de stocker temporairement les modifications et de les valider avant de les activer.

Dans Enterprise Manager, vous pouvez utiliser plusieurs pages de propriétés pour définir les directives du plan.

1. Dans la page General, associez les groupes de consommateurs de ressources aux plans. Indiquez la quantité de CPU allouée à chaque groupe ou sous-plan en indiquant une valeur MGMT_MTH.
2. Indiquez le degré de parallélisme maximal admis pour une opération dans un groupe de consommateurs de ressources.
3. Vous pouvez contrôler le nombre maximal de sessions actives simultanées autorisé dans un groupe de consommateurs de ressources. Chaque session entière d'exécution en parallèle est comptée en tant que session active.
4. Vous pouvez contrôler la quantité totale d'informations d'annulation pouvant être générée par un groupe de consommateurs de ressources.
5. Vous pouvez indiquer des valeurs de seuil, comme une limite relative à la durée d'exécution (en secondes), une limite d'E/S (en Mo) et une limite de demandes d'E/S (en nombre de demandes).
6. Vous pouvez indiquer la durée d'inactivité au terme de laquelle une session est arrêtée. Il est aussi possible de définir une durée maximale d'inactivité distincte pour les sessions qui bloquent d'autres sessions.

Remarque : Les diapositives qui suivent fournissent plus de détails sur les directives. Elles utilisent les numéros d'onglet indiqués dans la diapositive ci-dessus. Reportez-vous à cette dernière lorsque vous voyez la mention "Directive - Onglet n".

Méthodes d'allocation des ressources pour les plans d'allocation de ressources

Paramètre (commentaires)	Valeurs possibles
MGMT_MTH	EMPHASIS, RATIO
Allocation de CPU	
PARALLEL_DEGREE_LIMIT_MTH	PARALLEL_DEGREE_LIMIT_ABSOLUTE
Limitation du degré de parallélisme de n'importe quelle opération	
ACTIVE_SESS_POOL_MTH	PARALLEL_DEGREE_LIMIT_ABSOLUTE
Limitation du nombre de sessions actives, mise en file d'attente des sessions inactives	
QUEUING_MTH	FIFO_TIMEOUT
Contrôle des files d'attentes, mode d'entrée des sessions inactives dans le pool de sessions actives	

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Méthodes d'allocation des ressources pour les plans d'allocation de ressources

Les méthodes d'allocation des ressources déterminent comment Resource Manager alloue une ressource particulière à un groupe de consommateurs ou à un plan d'allocation de ressources. Lors de la création d'un plan d'allocation de ressources, vous définissez des valeurs pour les méthodes d'allocation des ressources.

Il existe deux moyens de définir la répartition de la CPU avec le paramètre MGMT_MTH :

- EMPHASIS est la méthode par défaut pour les plans à un seul niveau. Elle est aussi utilisée pour les plans à plusieurs niveaux qui utilisent des pourcentages pour indiquer la façon dont la CPU est distribuée entre les groupes de consommateurs de ressources.
- RATIO concerne les plans à un seul niveau qui utilisent des ratios pour définir la distribution de la CPU.

PARALLEL_DEGREE_LIMIT_MTH limite le degré de parallélisme pour toutes les opérations. Cette méthode ne peut être utilisée que pour les groupes de consommateurs de ressources. Elle ne s'applique pas aux sous-plans. La méthode PARALLEL_DEGREE_LIMIT_ABSOLUTE est la seule valeur possible. Elle indique le nombre de processus pouvant être affectés à une opération. Si plusieurs directives du plan d'allocation de ressources font référence au même sous-plan ou au même groupe de consommateurs de ressources, la limite utilisée pour le degré de parallélisme est le **minimum** de toutes les valeurs possibles.

Méthodes d'allocation des ressources pour les plans d'allocation de ressources (suite)

Le paramètre `ACTIVE_SESS_POOL_MTH` limite le nombre de sessions actives. Toutes les autres sessions sont inactives et patientent en file d'attente jusqu'à leur activation. La seule valeur (c'est-à-dire la seule méthode disponible) pour ce paramètre est `PARALLEL_DEGREE_LIMIT_ABSOLUTE`, soit sa valeur par défaut.

`QUEUING_MTH` contrôle l'ordre d'exécution des sessions inactives figurant dans la file d'attente. `FIFO_TIMEOUT` est la méthode par défaut et la seule disponible.

Comparaison entre EMPHASIS et RATIO

EMPHASIS	RATIO
La valeur indique le pourcentage maximal de ressources CPU qu'un groupe de consommateurs de ressources peut utiliser.	La valeur définit un nombre indiquant le ratio de ressources CPU à allouer au groupe de consommateurs de ressources.
Vous pouvez allouer des ressources pour un maximum de 8 niveaux différents.	Vous ne pouvez définir des valeurs que pour un niveau.
Le total des pourcentages d'un niveau donné doit être inférieur ou égal à 100.	Vous devez utiliser des valeurs entières, mais la somme des valeurs n'est pas limitée.
La valeur par défaut est NULL.	La valeur par défaut est NULL.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Comparaison entre EMPHASIS et RATIO

La méthode EMPHASIS d'allocation de la CPU permet de définir l'importance accordée aux sessions des différents groupes de consommateurs de ressources au sein d'un plan d'allocation de ressources. L'utilisation de la CPU est associée à des niveaux compris entre 1 et 8, 1 correspondant au niveau de priorité le plus élevé. Les pourcentages indiquent comment allouer la CPU à chaque groupe de consommateurs de ressources et à chaque niveau.

Les règles suivantes s'appliquent lors de l'allocation des ressources avec la méthode EMPHASIS :

- Les ressources CPU sont distribuées à un niveau donné en fonction des pourcentages indiqués. Le pourcentage de CPU défini pour un groupe de consommateurs de ressources correspond à un taux maximal pour le groupe à un niveau donné.
- Les ressources qui ne sont pas utilisées à un niveau donné sont mises à la disposition des groupes de consommateurs de ressources au niveau suivant. Par exemple, si les groupes de consommateurs de ressources au niveau 1 utilisent seulement 60 % des ressources disponibles, les 40 % qui restent sont mis à la disposition des groupes de consommateurs de ressources au niveau 2.
- Le total des pourcentages d'un niveau donné doit être inférieur ou égal à 100.
- Tous les niveaux pour lesquels aucune directive de plan d'allocation de ressources n'est explicitement définie utilisent par défaut 0 % de ressources pour tous les sous-plans ou groupes de consommateurs de ressources.
- La méthode EMPHASIS d'allocation des ressources évite les problèmes de manque de ressources, dans lesquels les consommateurs de priorité plus faible ne sont jamais autorisés à s'exécuter.

Comparaison entre EMPHASIS et RATIO (suite)

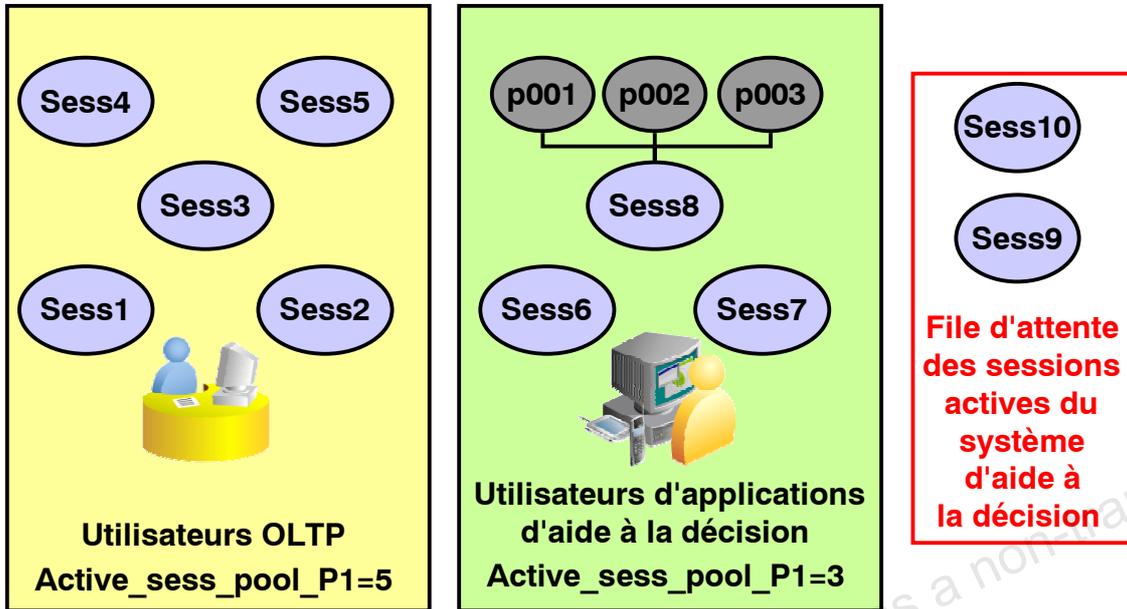
La stratégie RATIO est une méthode d'allocation de CPU à un seul niveau. Plutôt que des pourcentages, vous indiquez des valeurs correspondant au ratio de CPU que vous souhaitez accorder au groupe de consommateurs de ressources. Par exemple, si vous avez trois groupes de consommateurs de ressources nommés OLTP_USERS, DSS_USERS et BATCH_USERS, vous pouvez définir les ratios suivants :

- OLTP_USERS : 4
- DSS_USERS : 3
- BATCH_USERS : 2
- OTHER : 1

Cela revient à dire que les utilisateurs OLTP doivent bénéficier de 40 % des ressources disponibles, les utilisateurs DSS de 30 %, les utilisateurs BATCH de 20 % et tous les autres groupes de consommateurs de ressources de 10 %.

Si aucun consommateur des groupes OTHER et DSS_USERS n'utilise actuellement de ressources CPU, le groupe OLTP_USERS bénéficie de deux tiers des ressources disponibles et le groupe BATCH_USERS dispose du tiers restant.

Mécanisme du pool de sessions actives



Copyright © 2009, Oracle. Tous droits réservés.

ORACLE

Mécanisme du pool de sessions actives

L'utilisation d'un pool de sessions actives permet de contrôler le nombre maximal de sessions actives simultanées par groupe de consommateurs de ressources. Le DBA peut ainsi contrôler indirectement la quantité de ressources utilisée par un groupe de consommateurs puisque la consommation des ressources est proportionnelle au nombre de sessions actives. L'utilisation d'un pool de sessions actives peut permettre de réduire le nombre de serveurs faisant appel à des ressources du système. Cette réduction évite les opérations de pagination et de swap inefficaces, ainsi que d'autres surconsommations des ressources (par exemple la mémoire) provoquées par des tentatives d'exécution simultanée d'un nombre trop élevé de travaux.

Lorsque le pool est rempli de sessions actives, Resource Manager place en file d'attente toutes les sessions suivantes qui tentent de devenir actives, jusqu'à ce que d'autres sessions actives prennent fin ou deviennent inactives. Une session active est une session actuellement impliquée dans une transaction, une interrogation ou une opération en parallèle. Les processus enfant parallèles individuels ne sont pas comptabilisés en tant que sessions. L'ensemble de l'opération en parallèle compte pour une session active.

Il n'existe qu'une seule file d'attente par groupe de consommateurs de ressources et la méthode "premier entré, premier sorti" (FIFO) avec temporisation est appliquée. La file d'attente est implémentée en tant que structure mémoire et ne peut pas être interrogée directement.

Configurer le pool de sessions actives

Edit Resource Plan: DEFAULT_PLAN

Actions: Create Like Show SQL Revert Apply

General Parallelism **Session Pool** Undo Pool Threshold Idle Time

Specify a limit on the maximum number of concurrently active sessions for a consumer group. All other sessions will wait in an activation queue.

Group	Maximum Number of Active Sessions	Activation Queue Timeout (sec)
APPUSER	50 UNLIMITED	UNLIMITED
LOW_GROUP	UNLIMITED	UNLIMITED
ORA\$DIAGNOSTICS	UNLIMITED	UNLIMITED
OTHER_GROUPS	UNLIMITED	UNLIMITED
SYS_GROUP	UNLIMITED	UNLIMITED



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Configurer le pool de sessions actives

Vous pouvez facilement configurer les paramètres du pool de sessions actives d'un plan d'allocation de ressources à l'aide d'Enterprise Manager.

Par exemple, si vous limitez le nombre maximum de sessions actives à 50 pour le groupe de consommateurs APPUSER :

```
BEGIN
dbms_resource_manager.clear_pending_area();
dbms_resource_manager.create_pending_area();
dbms_resource_manager.update_plan_directive(
  plan => 'DEFAULT_PLAN',
  group_or_subplan => 'APPUSER',
  new_comment => '',
  new_active_sess_pool_p1 => 50,
  new_queueing_p1 => NULL,
  new_parallel_degree_limit_p1 => NULL,
  new_switch_group => '',
  new_switch_time => NULL,
  new_switch_estimate => false,
```

Configurer le pool de sessions actives (suite)

```
new_max_est_exec_time => NULL,  
new_undo_pool => NULL,  
new_max_idle_time => NULL,  
new_max_idle_blocker_time => NULL,  
mgmt_p1 => NULL,  
mgmt_p2 => NULL,  
mgmt_p3 => 60,  
mgmt_p4 => NULL,  
mgmt_p5 => NULL,  
mgmt_p6 => NULL,  
mgmt_p7 => NULL,  
mgmt_p8 => NULL,  
switch_io_megabytes => NULL,  
switch_io_reqs => NULL,  
switch_for call);  
dbms_resource_manager.submit_pending_area();  
END;
```

Définir des seuils

Indication d'une durée d'exécution limite :

- Estimation proactive de la durée d'exécution d'une opération (via les statistiques de l'optimiseur basé sur le coût). Valeur par défaut : UNLIMITED
- Indication de la durée d'exécution maximale estimée au niveau du groupe de consommateurs de ressources.
- Les travaux pour lesquels la durée estimée est supérieure à MAX_EST_EXEC_TIME : (ORA-07455) ne sont pas lancés.

Définition d'autres seuils :

- Limitation des E/S au niveau session avec SWITCH_IO_MEGABYTES (en Mo)
- Limitation des demandes d'E/S au niveau session avec SWITCH_IO_REQS

Retour au groupe de consommateurs d'origine avec SWITCH_FOR_CALL (Valeur par défaut : FALSE, le groupe de consommateurs n'est pas restauré)

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Définir des seuils

Vous pouvez définir le temps d'exécution maximal estimé d'une opération à un instant donné en configurant le paramètre MAX_EST_EXEC_TIME de la directive du plan d'allocation de ressources.

- Lorsque ce paramètre est défini, Database Resource Manager estime la durée d'un travail spécifique en utilisant les statistiques de l'optimiseur basé sur le coût.
- Si un groupe de consommateurs de ressources est associé à plusieurs directives du plan d'allocation, le paramètre MAX_EST_EXEC_TIME peut également être défini plusieurs fois. Database Resource Manager choisit alors la valeur la plus restrictive.
- Si cette estimation est supérieure à la valeur du paramètre MAX_EST_EXEC_TIME, l'opération n'est pas lancée et l'erreur ORA-07455 est générée. Ainsi, les travaux particulièrement volumineux qui sollicitent un trop grand nombre de ressources du système sont éliminés.
- La directive SWITCH_IO_MEGABYTES indique la quantité d'E/S (en Mo) qu'une session peut générer avant le déclenchement d'une action. La valeur par défaut est NULL, ce qui signifie une durée illimitée.
- La directive SWITCH_IO_REQS indique le nombre d'E/S qu'une session peut générer avant le déclenchement d'une action. La valeur par défaut est NULL, ce qui signifie une durée illimitée.
- La directive SWITCH_FOR_CALL indique que si une action est déclenchée à cause des paramètres SWITCH_TIME, SWITCH_IO_MEGABYTES ou SWITCH_IO_REQS, il y a un retour au groupe de consommateurs d'origine à la fin de l'appel principal. La valeur par défaut est FALSE, ce qui signifie que le groupe d'origine n'est pas rétabli à la fin de l'appel principal.

Définir des délais d'inactivité

Actions: Create Like Go Show SQL Revert Apply

General Parallelism Session Pool Undo Pool Threshold Idle Time

Specify the maximum time a session in the consumer group can be idle.

Group	Max Idle Time (sec)	Max Idle Time if Blocking Another Session (sec)
APPUSER	600	300
LOW_GROUP	UNLIMITED	UNLIMITED
ORA\$DIAGNOSTICS	UNLIMITED	UNLIMITED
OTHER_GROUP		
SYS_GROUP		

```
DBMS_RESOURCE_MANAGER.UPDATE_PLAN_DIRECTIVE
(PPLAN => 'DAY_PLAN',
 GROUP_OR_SUBPLAN => 'APPUSER',
 COMMENT => 'Limit Idle Time Example',
 NEW_MAX_IDLE_TIME => 600,
 NEW_MAX_IDLE_BLOCKER_TIME => 300);
```

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Définir des délais d'inactivité

Utilisez l'onglet Idle Time pour définir les délais d'inactivité maximum d'un plan d'allocation de ressources. Les options Max Idle Time (sec) et Max Idle Time if Blocking Another Session (sec) correspondent respectivement aux directives `NEW_MAX_IDLE_TIME` et `NEW_MAX_IDLE_BLOCKER_TIME` de la procédure `DBMS_RESOURCE_MANAGER.UPDATE_PLAN_DIRECTIVE`. Les deux délais sont exprimés en secondes.

La directive `NEW_MAX_IDLE_TIME` indique la durée maximale pendant laquelle une session n'est ni en cours d'exécution, ni en attente d'E/S. Lorsque la session reste inactive au-delà de cette limite, le processus PMON la ferme et initialise son statut. Vous pouvez également limiter la durée pendant laquelle une session inactive peut bloquer une autre session. Vous pouvez imposer cette limite en affectant à la directive de ressource `NEW_MAX_IDLE_BLOCKER_TIME` le nombre de secondes pendant lesquelles une session peut être inactive pendant le blocage d'une autre session. Vous pouvez également définir la valeur `UNLIMITED` afin d'indiquer qu'aucun délai maximum n'a été défini. La valeur par défaut est `NULL`, ce qui signifie une durée illimitée. Ces paramètres permettent un contrôle plus granulaire que les profils. Ceux-ci comprennent une valeur unique qui ne permet pas d'établir une distinction entre les sessions bloquantes et les sessions non bloquantes.

Dans l'exemple de la diapositive, le processus PMON ferme les sessions qui sont inactives pendant plus de 600 secondes. Il ferme également les sessions qui sont inactives pendant plus de 300 secondes et qui bloquent d'autres sessions. Le processus PMON effectue une vérification toutes les minutes. S'il trouve une session qui a dépassé l'une des limites, il la ferme et libère toutes les ressources associées.

Limiter l'utilisation de la CPU au niveau base de données

Conditions requises pour la consolidation au niveau base de données :

- Applications indépendantes les unes des autres
- Performances régulières

Les directives relatives à la CPU peuvent être utilisées pour les opérations suivantes :

- Définir une allocation de CPU minimum pour chaque application
- Indiquer comment les allocations inutilisées doivent être redistribuées
- Indiquer l'attribut `MAX_UTILIZATION_LIMIT` pour imposer une limite supérieure absolue relative à l'utilisation de la CPU (qui est prioritaire sur toute redistribution de CPU définie dans un plan)
- Candidat approprié : tâches de maintenance automatisées

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Limiter l'utilisation de la CPU au niveau base de données

Pour les sessions de base de données simultanées : la consolidation au niveau base de données requiert que les applications soient indépendantes l'une de l'autre. L'augmentation de la charge globale d'une application ne doit pas avoir d'impact sur les autres applications. En outre, les performances de chaque application doivent être régulières.

Stratégie fixe pour la gestion des ressources de CPU

L'attribut `MAX_UTILIZATION_LIMIT` associé aux directives de plan de ressources vous permet d'appliquer une limite supérieure absolue à la consommation de CPU d'un groupe de consommateurs. Cette limite absolue prévaut sur toute redistribution de CPU définie dans un plan.

Remarque : Les applications éligibles pour la consolidation au niveau base de données sont les tâches de maintenance automatisées, car elles peuvent utiliser jusqu'à 100 % des ressources CPU côté serveur. Vous pouvez définir une limite maximum pour chaque groupe de consommateurs exécutant des tâches automatiques.

Limiter l'utilisation de la CPU au niveau base de données

Définir des limites minimum et maximum pour l'utilisation de la CPU

<u>Plan de consolidation BdD No1</u>		
	Allocation CPU	Limite d'utilisation maximale
Appl 1	50%	60%
Appl 2	20%	30%
Appl 3	20%	30%
Appl 4	10%	20%

Définir uniquement des limites maximum d'utilisation de la CPU

<u>Plan de consolidation BdD No2</u>		
	Allocation CPU	Limite d'utilisation maximale
Appl 1	50%	60%
Appl 2	20%	60%
Appl 3	20%	60%
Appl 4	10%	60%

```
EXEC DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE( -
  plan                => 'db_consolidation_plan',
  group_or_subplan    => 'App_1',
  mgmt_pl             => 50,
  max_utilization_limit => 60);
```



Copyright © 2009, Oracle. Tous droits réservés.

Limiter l'utilisation de la CPU au niveau base de données (suite)

La directive `MAX_UTILIZATION_LIMIT` limite la consommation de CPU d'une application. Vous pouvez définir des limites minimum et maximum, comme illustré dans la diapositive ci-dessus.

L'exemple PL/SQL définit une valeur minimale en pourcentage (50%) pour l'allocation de ressources CPU de niveau 1 pour le groupe de consommateurs `APP_1`. Il définit aussi le pourcentage absolu (60%) autorisé pour ce groupe. Le plan utilisé est `DB_CONSOLIDATION_PLAN`.

Des commandes similaires peuvent être exécutées pour chaque groupe de consommateurs des exemples de tables.

Remarque : Dans les versions antérieures à Oracle Database 11gR2, la limite d'utilisation maximale implicite était fixée à 100%.

Limiter l'utilisation de CPU au niveau serveur : Mise en cage d'instance

- Gérer l'allocation de CPU sur un serveur multi-CPU avec plusieurs instances de base de données
- Activer la mise en cage d'instance :
 - Activer un plan d'allocation des ressources CPU

```
alter system set resource_manager_plan = 'default_plan';
```

- Définir le nombre maximal de CPU que l'instance peut utiliser à un instant quelconque

```
alter system set cpu_count=4;
```

Deux approches possibles :

- Allocation excédentaire : La somme des limites de CPU affectées aux différentes instances dépasse le nombre de CPU.
- Partitionnement : La somme des limites de CPU affectées aux différentes instances est égale au nombre de CPU.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Limiter l'utilisation de CPU au niveau serveur : Mise en cage d'instance

De nombreuses bases de données de test, de développement et de production à petite échelle n'utilisent pas toute la puissance des serveurs sur lesquels elles sont exécutées. Une alternative possible est alors la *consolidation au niveau serveur*. En exécutant plusieurs instances de bases de données sur le serveur, l'utilisation des ressources est mieux optimisée. Cependant, la surcharge d'une instance peut entraîner une contention au niveau de la CPU et avoir des incidences sur les autres bases.

La mise en cage d'instance est une méthode qui recourt au paramètre d'initialisation CPU_COUNT pour limiter le nombre de CPU qu'une instance peut utiliser. En outre, Oracle Database Resource Manager sert à allouer les CPU pour les sessions de base de données selon le plan des ressources d'instances.

La configuration de la mise en cage d'instance s'effectue en deux étapes :

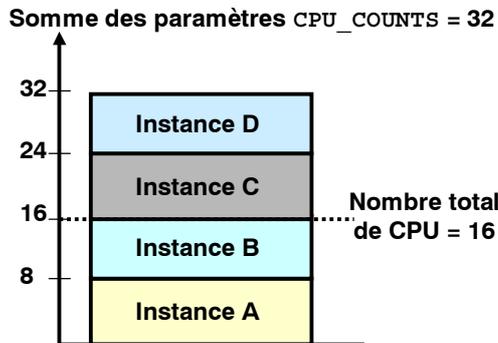
- Activez Resource Manager, qui limite la quantité de CPU consommée par l'instance de base de données.
- Activez le paramètre CPU_COUNT, qui indique le nombre maximum (limite) de CPU qu'une instance de base de données peut utiliser à tout moment.

Par défaut, le gestionnaire des ressources CPU suppose que l'instance de base de données peut utiliser toutes les CPU d'un serveur. Pour activer la mise en cage d'instance, il est possible d'utiliser n'importe quel plan de ressources comprenant des directives sur la CPU.

Exemples de mise en cage d'instance

Allocation excédentaire :

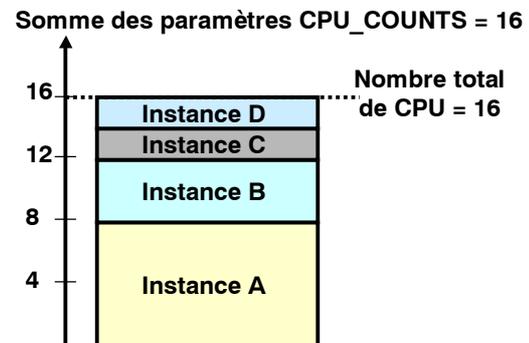
Une instance de base de données peut avoir un impact sur les autres.



Si les quatre instances sont actives, une instance peut obtenir $4 / (4 + 4 + 4 + 4) = 25\%$ de CPU.

Partitionnement :

Une instance de base de données n'a pas d'impact sur les autres.



Chaque instance possède un nombre dédié de CPU.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Exemples de mise en cage d'instance

Allocation excédentaire : Cette approche convient pour les bases de données non critiques dont la charge est faible, dans les systèmes de production qui n'ont pas un caractère sensible. Bien que les instances aient un impact les unes sur les autres, à un instant donné, il existe probablement une ou plusieurs instances inactives ou présentant une faible charge.

La mise en cage d'instance permet de limiter cet impact et d'obtenir des performances prévisibles. Dans l'exemple de gauche, où les quatre instances ont un paramètre CPU_COUNT défini avec la valeur 4, le pourcentage maximum de CPU qu'une base de données peut consommer à un instant quelconque est sa propre limite divisée par la somme des limites de toutes les bases de données actives. Dans cet exemple, une instance pourra consommer $4 / (4 + 4 + 4 + 4) = 25\%$ de la CPU. Si seulement deux instances sont actives, une instance pourra consommer $4 / (4 + 4) = 50\%$ de la CPU.

Partitionnement : Cette approche est recommandée pour les systèmes de production critiques. Elle empêche les instances d'interférer entre elles et fournit des performances prévisibles.

La mise en cage d'instance peut partitionner les ressources CPU en s'assurant que la somme des limites ne dépasse pas le nombre total de CPU. Dans l'exemple de droite, si quatre instances de base de données partagent un serveur contenant 16 CPU, leurs limites peuvent être définies comme suit : 8, 4, 2 et 2. Lorsque des ressources CPU dédiées sont affectées à une instance de base de données, le partitionnement fournit deux avantages :

- La charge CPU d'une instance de base de données ne peut pas affecter celle d'une autre instance.
- Les ressources CPU de chaque instance de base de données sont fixes. Cela permet d'obtenir des performances plus prévisibles.

Surveiller la mise en cage d'instance

Afficher la valeur du paramètre CPU_COUNT :

```
SELECT value FROM v$parameter WHERE name = 'cpu_count'  
AND (isdefault = 'FALSE' OR ismodified != 'FALSE');
```

Déterminer le statut de Resource Manager :

```
SELECT name FROM v$rsrc_plan  
WHERE is_top_plan = 'TRUE' AND cpu_managed = 'ON';
```

Gérer l'ajustement :

```
SELECT begin_time, consumer_group_name,  
       cpu_consumed_time, cpu_wait_time  
FROM v$rsrcmgrametric_history  
ORDER BY begin_time;
```

```
SELECT name, consumed_cpu_time, cpu_wait_time  
FROM v$rsrc_consumer_group;
```

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Surveiller la mise en cage d'instance

- Si le paramètre CPU_COUNT n'est pas défini, aucune valeur n'est renvoyée par la première interrogation.
- Si aucune ligne n'est renvoyée par la deuxième interrogation de la diapositive, Resource Manager ne gère pas la CPU. Si une ligne est renvoyée, elle indique le plan utilisé.

La mise en cage d'instance limite la consommation de CPU des processus en avant-plan en effectuant un ajustement (throttling). On dit qu'un processus en avant-plan est ajusté lorsqu'il attend l'événement Wait "resmgr:cpu quantum".

Vous pouvez surveiller le degré d'ajustement de deux manières :

- La vue V\$RSRCMGRAMETRIC_HISTORY indique la quantité de CPU consommée (CPU_CONSUMED_TIME) et le degré d'ajustement (CPU_WAIT_TIME) pour chaque minute de l'heure précédente. Les valeurs sont affichées en millisecondes.
- La vue V\$RSRC_CONSUMER_GROUP indique la quantité de CPU consommée et le degré d'ajustement depuis l'activation du gestionnaire de ressources CPU. Les valeurs sont affichées en millisecondes.

Remarque : Des études de cas sont présentées dans le livre blanc Oracle intitulé *Database Instance Caging: A Simple Approach to Server Consolidation*.

Mapping des groupes de consommateurs de ressources

The screenshot shows the 'Consumer Group Mappings' window with the 'Priorities' tab active. A table lists various mapping types with their current values and consumer groups. An 'Attribute Mappings' dialog box is open, showing a list of attributes that can be mapped to a consumer group.

Select	Priority	View	Value	Consumer
<input checked="" type="radio"/>	1	Service Module and Action	No Mappings Specified	
<input type="radio"/>	2	Service and Module	No Mappings Specified	
<input type="radio"/>	3	Module and Action	No Mappings Specified	
<input type="radio"/>	4	Module	No Mappings Specified	
<input type="radio"/>	5	Service	No Mappings Specified	
<input type="radio"/>	6	Oracle User	HR SCOTT SYS, SYSTEM	APPUSER LOW_GRO SYS_GROU
<input type="radio"/>	7	Client Program	No Mappings Specified	
<input type="radio"/>	8	Client OS User	ORACLE	SYS_GROU
<input type="radio"/>	9	Client Machine	No Mappings Specified	

Attribute Mappings

- Service Module and Action
- Service and Module
- Module and Action
- Module
- Service
- Oracle User
- Client Program
- Client OS User
- Client Machine

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Mapping des groupes de consommateurs de ressources

Vous pouvez configurer Database Resource Manager de manière à ce qu'il affecte automatiquement des groupes de consommateurs de ressources aux sessions en définissant des mappings entre les attributs de session et les groupes. En outre, vous pouvez classer les mappings afin d'indiquer la priorité à respecter en cas de conflit. Il existe deux types d'attribut de session : les attributs de connexion et les attributs d'exécution. Les attributs de connexion (les cinq derniers attributs de la liste Attribute Mappings présentée dans la diapositive ci-dessus) ne sont utilisés qu'au moment de l'ouverture d'une session, lorsque Database Resource Manager détermine le groupe de consommateurs de ressources initial de la session. A l'inverse, une session qui s'est déjà connectée peut ultérieurement être affectée à un autre groupe de consommateurs de ressources sur la base de ses attributs d'exécution.

Dans la page d'accueil de Database Control, cliquez sur l'onglet Server, puis sur le lien Resource Consumer Group Mappings, dans la section Resource Manager. Pour chaque attribut, définissez un mapping qui détermine la façon d'identifier une session (par exemple, nom utilisateur) et un groupe de consommateurs de ressources. Ajoutez ou supprimez des lignes pour chacune des catégories de groupe de consommateurs de ressources, en fonction des besoins, et entrez le texte qui identifie l'utilisateur, le client, le module ou le service dans le groupe correspondant. Vous pouvez établir une priorité entre les mappings d'attributs pour la résolution des conflits, via l'onglet Priorities. Classez les mappings du plus important au moins important à l'aide des flèches de navigation, comme indiqué dans la diapositive. Les mappings figurant dans la partie supérieure de la liste ont la priorité la plus élevée.

Mapping des groupes de consommateurs de ressources (suite)

Dans EM Database Control, vous pouvez facilement afficher le code SQL généré par vos actions en cliquant sur le bouton Show SQL.

Exemple pour attribuer à l'utilisateur du système d'exploitation client une priorité plus élevée que le programme client :

```
BEGIN
dbms_resource_manager.clear_pending_area();
dbms_resource_manager.create_pending_area();
dbms_resource_manager.set_consumer_group_mapping(
    dbms_resource_manager.oracle_user,
    'SCOTT',
    'LOW_GROUP'
);
dbms_resource_manager.set_consumer_group_mapping_pri(
    EXPLICIT => 1, SERVICE_MODULE_ACTION => 2,
    SERVICE_MODULE => 3,
    MODULE_NAME_ACTION => 4,
    MODULE_NAME => 5,
    SERVICE_NAME => 6,
    ORACLE_USER => 7,
    CLIENT_OS_USER => 8,
    CLIENT_PROGRAM => 9,
    CLIENT_MACHINE => 10
);
dbms_resource_manager.submit_pending_area();
END;
```

Activer un plan d'allocation de ressources

Select	Plan	Status	Description	Scheduler Windows
<input checked="" type="radio"/>	DEFAULT MAINTENANCE PLAN		Default plan for maintenance windows that prioritizes SYS_GROUP operations and allocates the remaining 5% to diagnostic operations and 25% to automated maintenance operations.	MONDAY WINDOW TUESDAY WINDOW WEDNESDAY WINDOW THURSDAY WINDOW FRIDAY WINDOW SUNDAY WINDOW SATURDAY WINDOW

EM > Server > Settings (dans la section Resource Manager)

ORACLE Enterprise Manager 11g Database Control

Database Instance: orcl > Logged in As SYS

Resource Manager Settings

Active Resource Plan **No Active Plan**

Available Resource Plans **MIXED_WORKLOAD_PLAN**

Configure properties of Resource Management that apply to all Resource Plans.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Activer un plan d'allocation de ressources

Vous pouvez utiliser la page Plans d'Enterprise Manager pour gérer les plans d'allocation de ressources. Pour activer un plan, sélectionnez-le, choisissez Activate dans la liste déroulante Actions, puis cliquez sur Go. Le plan que vous avez sélectionné devient alors le plan de premier niveau actuel de l'instance.

Utiliser le paramètre d'initialisation RESOURCE_MANAGER_PLAN

Le plan d'une instance est défini via le paramètre d'initialisation de base de données RESOURCE_MANAGER_PLAN. Celui-ci permet d'indiquer le plan de premier niveau à associer à l'instance. Si aucun plan n'est défini, Resource Manager demeure inactif pour cette instance.

Vous pouvez activer, désactiver ou modifier le plan de premier niveau actuel à l'aide de l'instruction ALTER SYSTEM. Lorsqu'un plan d'allocation de ressources est modifié à l'aide de cette commande, la modification prend effet instantanément.

Si le paramètre est défini dans un fichier de paramètres et que le plan désigné n'est pas défini dans la base de données, cette dernière ne peut pas être ouverte avec ce fichier de paramètres. L'erreur suivante est renvoyée :

```
ORA-07452: specified resource manager plan does not exist
in the data dictionary
```

Si cette erreur est rencontrée, le paramètre doit être modifié afin d'afficher une valeur correcte avant tout redémarrage de l'instance.

Informations relatives à Database Resource Manager

Nom de la vue	Informations
DBA_RSRC_PLANS	Plans et statut
DBA_RSRC_PLAN_DIRECTIVES	Directives de plan
DBA_RSRC_CONSUMER_GROUPS	Groupes de consommateurs de ressources
DBA_RSRC_CONSUMER_GROUP_PRIVS	Utilisateurs/rôles
DBA_RSRC_GROUP_MAPPINGS	Mapping de groupes de consommateurs de ressources
DBA_RSRC_MAPPING_PRIORITY	Priorité du mapping
DBA_USERS	Colonne initial_rsrc_consumer_group
DBA_RSRC_MANAGER_SYSTEM_PRIVS	Utilisateurs/rôles

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Informations relatives à Database Resource Manager

Le dictionnaire de données offre plusieurs vues permettant de vérifier les plans d'allocation de ressources, les groupes de consommateurs de ressources et les directives de plan déclarés dans une instance. Cette section présente des informations utiles pouvant être obtenues à partir de ces vues. Pour des informations plus détaillées sur le contenu de chacune de ces vues, reportez-vous au manuel *Oracle Database Reference*.

Utilisez l'interrogation suivante pour obtenir des informations sur les plans d'allocation de ressources définis dans la base de données :

```
SQL> SELECT plan, num_plan_directives, status, mandatory
       2 FROM dba_rsrc_plans;
PLAN          NUM_PLAN_DIRECTIVES STATUS      MAN
-----
DEFAULT_PLAN          3 ACTIVE     NO
INTERNAL QUIESCE      2 ACTIVE     YES
INTERNAL_PLAN         1 ACTIVE     YES
BUGDB_PLAN           4 ACTIVE     NO
MAILDB_PLAN          3 ACTIVE     NO
MYDB_PLAN            3 ACTIVE     NO
```

Le statut ACTIVE indique que le plan a été soumis et qu'il peut être utilisé, tandis que le statut PENDING indique que le plan a été créé, mais qu'il se trouve encore en zone d'attente.

Si la valeur YES est affectée à la colonne mandatory, le plan ne peut pas être supprimé.

Surveiller Resource Manager

Queued Sessions						
Consumer Group	Configured Limits		Statistics			
	Active Session Limit	Maximum Time in Queue (sec)	Current Queued Sessions	Total Queued Sessions	Time in Queue (sec)	Queue Time Outs
APPUSER	UNLIMITED	UNLIMITED	0	0	0	0
SYS_GROUP	UNLIMITED	UNLIMITED	0	0	0	0
OTHER_GROUPS	UNLIMITED	UNLIMITED	0	0	0	0
ORA\$DIAGNOSTICS	UNLIMITED	UNLIMITED	0	0	0	0
LOW_GROUP	UNLIMITED	UNLIMITED	0	0	0	0

Automatic Reprioritization							
Consumer Group	Configured Limits			Statistics			
	CPU Time Limit (ms)	I/O Requests	I/O (MB)	Switch into Group	Switch out of the Group	Active Sessions Killed	Calls Cancelled
APPUSER	UNLIMITED	UNLIMITED	UNLIMITED	0	0	0	0
SYS_GROUP	UNLIMITED	UNLIMITED	UNLIMITED	0	0	0	0
OTHER_GROUPS	UNLIMITED	UNLIMITED	UNLIMITED	0	0	0	0
ORA\$DIAGNOSTICS	UNLIMITED	UNLIMITED	UNLIMITED	0	0	0	0
LOW_GROUP	UNLIMITED	UNLIMITED	UNLIMITED	0	0	0	0

Idle				
Consumer Group	Idle Blocker Limit		Idle Limit	
	Maximum Idle Blocker Time (sec)	Blocker Sessions Killed	Maximum Idle Time (sec)	Idle Sessions Killed
APPUSER	UNLIMITED	0	UNLIMITED	0
SYS_GROUP	UNLIMITED	0	UNLIMITED	0
OTHER_GROUPS	UNLIMITED	0	UNLIMITED	0
ORA\$DIAGNOSTICS	UNLIMITED	0	UNLIMITED	0
LOW_GROUP	UNLIMITED	0	UNLIMITED	0

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Surveiller Resource Manager

Vous pouvez surveiller le fonctionnement de Database Resource Manager au niveau session. Cette surveillance est possible grâce à l'intégration du moniteur ADDM (Automatic Database Diagnostic Monitor).

Il existe différents moyens de gérer et de surveiller Resource Manager via EM Database Control. Dans l'onglet Server, cliquez sur le lien Statistics dans la section Resource Manager.

La page Resource Manager Statistics affiche un groupe de statistiques et de graphiques qui illustrent l'état actuel du plan d'allocation de ressources actif. Vous pouvez consulter les statistiques concernant le plan actuellement actif.

Concernant l'utilisation des ressources, vous pouvez consulter la CPU consommée, le nombre de demandes d'E/S par seconde et le nombre de mégaoctets d'E:S exécutés par seconde. Un autre graphique affiche Resource Manager Induced Waits. Enfin, vous trouvez des statistiques concernant Queued Sessions, Automatic Reprioritization et le temps d'inactivité.

Surveiller Resource Manager

- V\$SESSION : contient la colonne `resource_consumer_group` dans laquelle apparaît le groupe actuel d'une session
- V\$RSRC_PLAN : vue dans laquelle apparaît le plan d'allocation de ressources actif
- V\$RSRC_CONSUMER_GROUP : vue contenant les statistiques relatives à tous les groupes actifs

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Surveiller Resource Manager (suite)

Utilisation de la CPU

Le système contient au moins trois vues qui fournissent des informations sur l'utilisation de la CPU dans la base de données Oracle :

- La vue V\$RSRC_CONSUMER_GROUP affiche les statistiques relatives à l'utilisation de la CPU par groupe de consommateurs de ressources, si Oracle Database Resource Manager est en cours d'exécution. Cette vue affiche les données relatives aux groupes de consommateurs de ressource actuellement actifs.
- La vue V\$SYSSTAT affiche l'utilisation de la CPU dans la base de données Oracle pour toutes les sessions. La statistique "CPU used by this session" affiche la CPU totale utilisée par toutes les sessions.
- La vue V\$SESSTAT affiche l'utilisation de la CPU dans la base de données Oracle par session. Vous pouvez l'utiliser pour connaître la session qui utilise le plus la CPU.

Vue V\$RSRC_CONSUMER_GROUP

Voici une brève description de certaines colonnes de cette vue.

- **name** : nom du groupe de consommateurs de ressources.
- **active_sessions** : nombre de sessions actuellement actives dans ce groupe de consommateurs de ressources.
- **execution_waiters** : nombre de sessions actives en attente d'une tranche de temps.
- **requests** : nombre cumulé de demandes exécutées dans ce groupe de consommateurs de ressources.
- **cpu_wait_time** : durée cumulée pendant laquelle les sessions ont attendu la CPU.
- **consumed_cpu_time** : temps CPU cumulé consommé par l'ensemble des sessions.

Surveiller Resource Manager (suite)

Aucune vue n'affiche directement la file d'attente du pool de sessions actives, mais vous pouvez obtenir des informations via les vues suivantes :

- **V\$SESSION** : La colonne `current_queue_duration` indique la durée de la présence en file d'attente d'une session. La durée est de 0 (zéro) pour les sessions qui ne sont actuellement pas en file d'attente.
- **V\$RSRC_CONSUMER_GROUP** : La colonne `queue_length` indique le nombre de sessions actuellement en file d'attente par groupe de consommateurs de ressources.

Quiz

Sélectionnez toutes les affirmations qui sont vraies à propos de Resource manager et de ses fonctionnalités :

1. Vous pouvez indiquer une valeur de seuil pour la durée d'exécution, mais pas pour les E/S au niveau session.
2. Vous pouvez limiter l'utilisation de la CPU au niveau base de données pour rendre les applications indépendantes les unes des autres.
3. Sur un serveur multi-CPU comprenant plusieurs instances de base de données, vous pouvez limiter l'utilisation de chaque CPU en activant la mise en cage d'instance.
4. Lorsque les paramètres SWITCH_TIME, SWITCH_IO_MEGABYTES ou SWITCH_IO_REQS entraînent un changement de groupe de consommateurs de ressources, il n'est plus possible de rétablir le groupe d'origine.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Réponses : 2, 3

Synthèse

Ce chapitre vous a permis d'apprendre à :

- configurer Database Resource Manager
- créer des plans d'allocation de ressources et y accéder
- créer des groupes de consommateurs de ressources
- définir des directives pour l'allocation de ressources à des groupes de consommateurs
- associer des groupes de consommateurs de ressources à des plans
- activer un plan d'allocation de ressources
- surveiller Resource Manager

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Présentation de l'exercice 16 : Utiliser Resource Manager

Cet exercice porte sur les points suivants :

- Créer un groupe de consommateurs de ressources
- Définir des directives d'allocation des ressources CPU pour les groupes de consommateurs de ressources
- Associer des utilisateurs à un groupe de consommateurs de ressources
- Activer un plan d'allocation de ressources
- Effectuer des tests dans SQL*Plus
- Désactiver un plan d'allocation de ressources

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

17

Automatiser des tâches avec le planificateur

Laurent PERPETUE (laurent.perpetue@sufr.fr) has a non-transferable license to use this Student Guide

Objectifs

A la fin de ce chapitre, vous pourrez :

- simplifier les tâches de gestion à l'aide du planificateur
- créer un travail, un programme et une planification
- surveiller l'exécution d'un travail
- utiliser une planification basée sur une date/heure ou sur les événements pour exécuter les travaux du planificateur
- décrire l'utilisation des fenêtres, des groupes de fenêtres, des classes de travaux et des groupes de consommateurs
- utiliser la notification par e-mail
- utiliser des chaînes de travaux pour effectuer une série de tâches interdépendantes
- décrire les travaux du planificateur sur les systèmes distants
- utiliser les fonctionnalités avancées du planificateur pour définir la priorité des travaux

ORACLE

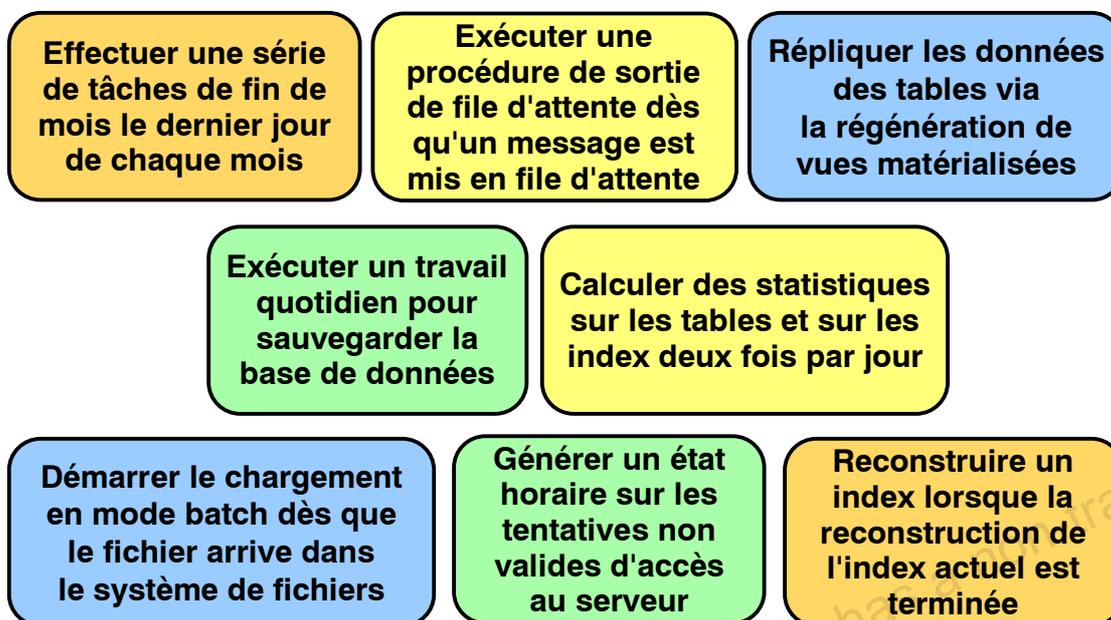
Copyright © 2009, Oracle. Tous droits réservés.

Objectifs

Pour plus d'informations sur les différents composants du planificateur et leurs interactions, reportez-vous au manuel *Oracle Database Administrator's Guide*.

Pour obtenir des informations détaillées sur le package DBMS_SCHEDULER, reportez-vous au manuel *Oracle Database PL/SQL Packages and Types Reference*.

Simplifier les tâches de gestion



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Simplifier les tâches de gestion

Dans l'environnement Oracle, de nombreuses tâches nécessitent des fonctionnalités de planification des travaux. La maintenance de routine de la base de données et la logique applicative nécessitent la planification et l'exécution périodique de travaux.

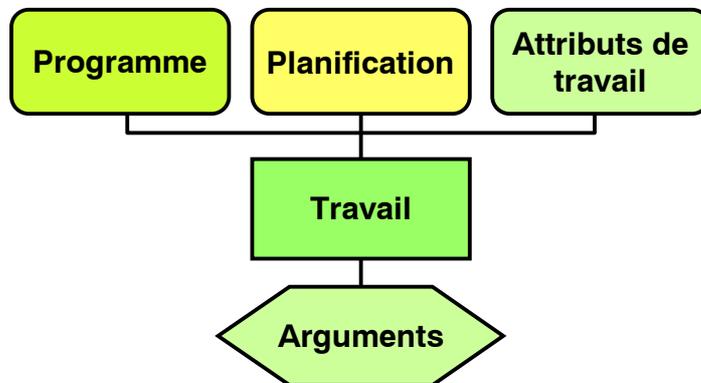
Les applications B-to-B requièrent la planification d'événements métier. Les DBA doivent planifier des travaux de maintenance réguliers au cours de périodes spécifiques.

La base de données Oracle fournit des fonctionnalités de planification avancées par le biais de son planificateur, qui est un ensemble de fonctions et de procédures contenues dans le package DBMS_SCHEDULER. Le planificateur peut être appelé dans n'importe quel environnement SQL ou via Enterprise Manager (EM).

Le planificateur permet aux DBA et aux développeurs d'applications de contrôler le moment d'exécution des différentes tâches dans l'environnement de base de données, et les modalités d'exécution. Ces tâches peuvent être longues et complexes. Le planificateur vous aide à les gérer et à les planifier.

Le planificateur peut lancer des travaux à un instant donné ou lorsqu'un événement spécifique se produit. Il peut aussi générer des événements lorsqu'un travail change de statut (par exemple, lorsqu'il passe de RUNNING à COMPLETE). Vous pouvez également utiliser une série nommée de programmes qui sont associés pour répondre à un objectif particulier.

Composants essentiels



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Composants essentiels

Un travail comprend deux composants obligatoire : l'action devant être effectuée et planifiée (quoi) et le moment où cette action doit avoir lieu (quand). La composante "quoi" est définie dans la partie commande et par les attributs du travail : paramètres `job_type` et `job_action`. La composante "quand" est indiquée via une "planification", qui peut être basée sur une date/heure ou sur des événements, ou bien dépendre du résultat d'autres travaux.

Le planificateur utilise trois composants principaux :

- Un **travail** définit ce qui doit être exécuté. Par exemple, le composant à exécuter peut être une procédure PL/SQL, un exécutable binaire natif, une application Java ou un script shell. Vous pouvez préciser le programme (quoi) et la planification (quand) dans le cadre de la définition du travail, ou bien utiliser un programme ou une planification qui existent déjà. Il est possible de recourir à des arguments pour personnaliser le comportement que doit avoir le travail lors de l'exécution.
- Une **planification** définit le moment où un travail est exécuté et le nombre de répétitions de l'exécution. Elle peut être basée sur une date/heure ou sur un événement. Vous pouvez définir la planification associée à un travail en utilisant une série de dates/heures, un événement, ou une combinaison des deux, ainsi que des spécifications supplémentaires précisant l'intervalle de répétition. La planification d'un travail peut être stockée séparément. Vous pouvez ainsi l'utiliser pour d'autres travaux.
- Un **programme** est un ensemble de métadonnées concernant un exécutable, un script ou une procédure. Un travail automatisé exécute une tâche. En utilisant un programme, vous pouvez modifier la partie tâche d'un travail (partie "quoi") sans modifier le travail proprement dit. Vous pouvez définir des arguments pour un programme, ce qui permet aux utilisateurs de modifier le comportement de la tâche lors de l'exécution.

Workflow de base

Pour simplifier les tâches de gestion dans le planificateur :

1. Créez un programme (activé ou désactivé)—facultatif
 - Pour réutiliser cette action dans plusieurs travaux
 - Pour modifier la planification d'un travail sans avoir à recréer le bloc PL/SQL
2. Créez et utilisez une planification.
3. Créez et soumettez un travail.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Workflow de base

Vous pouvez réaliser les différentes étapes dans l'environnement graphique d'Enterprise Manager ou utiliser le package PL/SQL DBMS_SCHEDULER via la ligne de commande.

1. Créer un programme

Pour créer un programme, utilisez la procédure CREATE_PROGRAM. Toutefois, la création d'un programme ne s'effectue pas nécessairement à l'aide du planificateur. Vous pouvez également encoder l'action à exécuter dans un bloc PL/SQL anonyme au sein de la procédure CREATE_JOB. Lorsque vous créez un programme à part, vous définissez l'action correspondante une fois, puis vous pouvez la réutiliser dans d'autres travaux. Cela vous permet de modifier la planification d'un travail sans avoir à recréer le bloc PL/SQL.

Par défaut, le programme créé est désactivé (sauf si vous avez attribué la valeur TRUE au paramètre enabled). Un programme désactivé ne peut pas être exécuté par un travail. Il doit d'abord être activé. Pour indiquer qu'un programme doit être créé avec le statut Enabled, affectez la valeur TRUE au paramètre enabled.

2. Créer et utiliser des planifications

La planification associée à un travail peut être prédéfinie (créée avec la procédure DBMS_SCHEDULER.CREATE_SCHEDULE) ou définie lors de la création du travail.

Workflow de base (suite)

Créer et utiliser des planifications (suite)

La planification comprend des attributs qui précisent le moment où le travail doit être exécuté. Par exemple :

- une heure de début, qui indique à quel moment le travail va être sélectionné en vue de son exécution, et une heure de fin, qui indique l'heure après laquelle le travail n'est plus valide et n'est plus planifié,
- une expression indiquant l'intervalle de répétition du travail,
- une planification complexe combinant plusieurs planifications existantes,
- une condition ou un changement de statut (événement) qui doit survenir pour que le travail soit lancé.

Lorsque vous utilisez une planification (au lieu d'indiquer les heures d'exécution d'un travail dans la définition de celui-ci), vous pouvez gérer l'exécution planifiée de plusieurs travaux sans avoir à mettre à jour plusieurs définitions de travaux. Si une planification est modifiée, chaque travail qui l'utilise prend automatiquement en compte sa nouvelle définition.

3. Créer et exécuter un travail

Un travail comprend une planification, la description des actions à effectuer et les arguments supplémentaires éventuellement requis. Vous pouvez définir de nombreux attributs pour un travail. Les attributs contrôlent le mode d'exécution du travail.

Quiz

Sélectionnez les affirmations qui sont vraies concernant le planificateur :

1. L'utilisation du planificateur implique la création d'un programme.
2. Lorsque l'action associée à un travail est définie dans un programme (plutôt que directement dans le travail), vous pouvez modifier la planification du travail sans avoir à recréer le bloc PL/SQL.
3. Il est possible de créer un travail de manière facultative lors de l'utilisation du planificateur.
4. Chaque travail doit être associé à une planification. Celle-ci peut être prédéfinie ou être définie lors de la création du travail.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Réponses : 2, 4

Travaux légers persistants

Les travaux légers persistants :

- réduisent la surcharge et la durée nécessaires au démarrage d'un travail
- présentent un faible encombrement sur le disque pour les métadonnées de travail et pour le stockage des données d'exécution
- sont créés à partir d'un modèle de travail (depuis la ligne de commande)

```
BEGIN
DBMS_SCHEDULER.CREATE_JOB (
  job_name           => 'my_lightweight_job2',
  program_name       => 'MY_PROG',
  schedule_name      => 'MY_SCHED',
  job_style           => 'LIGHTWEIGHT');
END;
```

/

Choisissez le travail approprié :

- Utilisez des travaux standard pour une flexibilité maximale.
- Utilisez des travaux légers persistants lorsque vous devez créer un grand nombre de travaux dans un délai très court.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Travaux légers persistants

Caractéristiques des travaux légers :

- Ils sont destinés aux clients qui ont besoin de créer des centaines de travaux par seconde. Un travail standard crée un objet de base de données qui le décrit, qui modifie plusieurs tables et qui crée des informations de journalisation dans le processus. La surcharge associée aux besoins de ce type de travail est importante. Dans le planificateur de la base de données Oracle figure un *travail léger (lightweight job) persistant*. L'objectif d'un travail léger est de réduire la surcharge et la durée nécessaires au lancement d'un travail. Une quantité minimale de métadonnées est créée pour le travail. La durée nécessaire au lancement d'un travail et les informations de journalisation créées à cette occasion sont ainsi réduites.
- Ils occupent peu d'espace sur le disque pour les métadonnées de travail et pour le stockage des données d'exécution. Ce faible encombrement permet également un équilibrage de la charge dans les environnements RAC.
- Ils sont toujours créés à partir d'un modèle de travail. Ce dernier peut être une procédure stockée ou un programme. La procédure stockée contient l'ensemble des informations nécessaires au travail, y compris les privilèges. Quelques attributs peuvent être associés au travail : arguments et planification.
- Ils doivent être créés depuis la ligne de commande. L'argument JOB_STYLE n'est pas disponible dans EM.

Dans l'exemple, MY_PROG est le modèle de travail et la planification est appliquée à partir d'une planification nommée.

Utiliser une planification basée sur une date/heure ou sur les événements

ORACLE Enterprise Manager 11g Database Control

Database Instance: orcl > Scheduler Jobs > Create Job

General Schedule Options

Schedule Type: Standard

Time Zone: Standard Using PL/SQL for repeated interval

Repeat: Do Not Repeat

Start: Immed

Time: 6:20:00 AM

Planification

Temps
- Expression de planification
- Expression de type date-heure

Evénement

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Utiliser une planification basée sur une date/heure ou sur les événements

Pour définir une planification basée sur une date/heure pour un travail, vous pouvez indiquer une expression de planification ou une expression date-heure. Lors de l'utilisation d'une expression de planification, l'heure de début suivante d'un travail est calculée à partir de l'intervalle de répétition et de la date de début du travail. Lors de l'utilisation d'une expression date-heure, l'expression indiquée détermine l'heure de la prochaine exécution du travail. Si aucun intervalle de répétition n'est indiqué, le travail s'exécute une seule fois à la date de début indiquée.

Si un travail utilise une planification basée sur les événements, il s'exécute lorsque l'événement associé est détecté. D'un point de vue très général, un événement peut être considéré comme un changement de statut. Un événement se produit lorsqu'une condition booléenne passe du statut FALSE à TRUE, ou du statut TRUE à FALSE.

Le planificateur utilise Oracle Streams Advanced Queuing (AQ) pour générer et consommer des événements.

Remarque : Le planificateur ne peut pas garantir qu'un travail s'exécutera exactement à l'heure prévue car il se peut que le système soit surchargé et que des ressources soient indisponibles.

Créer un travail basé sur une date/heure

Exemple : Créer un travail qui appelle un script de sauvegarde chaque nuit à 23:00, à partir de ce soir.

```
BEGIN
  DBMS_SCHEDULER.CREATE_JOB (
    job_name=>'HR.DO_BACKUP',
    job_type => 'EXECUTABLE',
    job_action =>
      '/home/usr/dba/rman/nightly_incr.sh',
    start_date=> SYSDATE,
    repeat_interval=>'FREQ=DAILY;BYHOUR=23',
      /* next night at 11:00 PM */
    comments => 'Nightly incremental backups');
END;
/
```



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Créer un travail basé sur une date/heure

Utilisez la procédure `CREATE_JOB` du package `DBMS_SCHEDULER` pour créer un travail. Par défaut, les travaux créés sont désactivés. Ils ne deviennent actifs et planifiés que lorsqu'ils sont activés de manière explicite. Tous les travaux portent un nom de la forme : `[schema.] name`.

Pour que l'heure d'exécution du travail soit ajustée automatiquement lors du passage à l'heure d'été, utilisez le paramètre `SYSTIMESTAMP` et indiquez un fuseau horaire.

Par défaut, un travail est créé dans le schéma actuel. Vous pouvez créer un travail dans un autre schéma en indiquant le nom de ce dernier, comme le montre l'exemple de la diapositive ci-dessus. Le propriétaire d'un travail est l'utilisateur dans le schéma duquel le travail est créé. Le créateur du travail est l'utilisateur qui effectue la création. Les travaux sont exécutés avec les privilèges du propriétaire du travail. L'environnement NLS (National Language Support) dans lequel s'exécute le travail est celui présent lors de la création du travail. Le paramètre `job_type` indique le type de tâche devant être effectué par le travail. Les valeurs possibles sont les suivantes :

- **PLSQL_BLOCK** : bloc PL/SQL anonyme.
- **STORED_PROCEDURE** : procédure PL/SQL, Java ou externe nommée.
- **EXECUTABLE** : commande pouvant être exécutée à partir de la ligne de commande du système d'exploitation.

Créer un travail basé sur une date/heure (suite)

La valeur du paramètre `job_action` peut être le nom de la procédure à exécuter, le nom d'un script ou d'une commande du système d'exploitation, ou un bloc de code PL/SQL anonyme, en fonction de la valeur du paramètre `job_type`.

Dans l'exemple de la diapositive, `job_type` a la valeur `EXECUTABLE` et `job_action` indique le chemin complet dépendant du système d'exploitation de l'exécutable externe souhaité plus tout argument de ligne de commande éventuel.

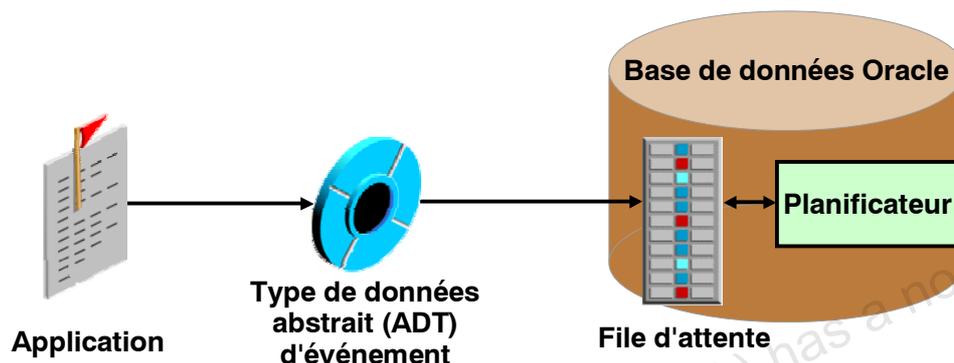
Un travail externe est un travail qui s'exécute en dehors de la base de données. Tous les travaux externes sont exécutés pour le compte d'un utilisateur `GUEST` ayant peu de privilèges, conformément à ce qui a été défini par le `DBA` lors de la configuration de ces travaux. Etant donné que l'exécutable est associé à un compte `GUEST` ayant peu de privilèges, vous devez vérifier que ce dernier a accès aux fichiers et aux ressources nécessaires. Toutes les plates-formes prennent en charge les travaux externes, à quelques exceptions près. Lorsque ce n'est pas le cas, le fait de créer ou de définir l'attribut d'un travail ou d'un programme avec le type `EXECUTABLE` renvoie une erreur.

Reportez-vous à la documentation propre à votre plate-forme Oracle Database pour plus d'informations sur la façon de configurer l'environnement en vue d'exécuter des programmes externes avec le planificateur.

Créer une planification basée sur les événements

Pour créer un travail basé sur les événements, vous devez définir :

- la spécification d'une file d'attente (dans laquelle l'application place les messages demandant le démarrage d'un travail)
- une condition d'événement (même syntaxe qu'une condition de règle Oracle Streams AQ) qui lance le travail lorsqu'elle a la valeur `TRUE`



Copyright © 2009, Oracle. Tous droits réservés.

ORACLE

Créer une planification basée sur les événements

Les travaux peuvent être déclenchés par des événements. Une application peut demander au planificateur de lancer un travail en plaçant un message dans une file d'attente Oracle Streams. Un travail lancé selon cette méthode est appelé un travail basé sur les événements. Pour créer un tel travail, vous devez définir les deux attributs supplémentaires suivants avec la procédure `CREATE_JOB` :

- **queue_spec** : Spécification de file d'attente, qui inclut le nom de la file d'attente dans laquelle l'application place des messages pour générer des événements de lancement de travail ou, dans le cas d'une file d'attente sécurisée, une paire `<queue_name>`, `<agent_name>`.
- **event_condition** : Expression conditionnelle basée sur des propriétés de message qui doivent prendre la valeur `TRUE` pour que le message entraîne le lancement du travail. Vous pouvez inclure des propriétés de données utilisateur dans l'expression, à condition que les données du message soient d'un type Objet défini par l'utilisateur et que vous ajoutiez le préfixe `tab.user_data` aux attributs d'objet utilisés dans l'expression.

Vous pouvez soit indiquer les attributs `queue_spec` et `event_condition` directement dans le code d'appel de procédure, soit créer une planification basée sur les événements mentionnant ces attributs, puis créer un travail qui référence cette planification.

Créer des planifications basées sur les événements avec Enterprise Manager

The screenshot shows the 'Schedule' configuration interface in Oracle Enterprise Manager. It is divided into two main sections: 'Schedule' and 'Event Parameters'.
In the 'Schedule' section, the 'Time Zone' is set to 'America/New_York' and the 'Schedule Type' is set to 'Event'.
In the 'Event Parameters' section, the 'Queue Name' is 'SYS.ALERT_QUE', the 'Agent Name' is 'ADMIN_AGNT1', and the 'Condition' is 'tab.user_data.event_type = "DISK_FULL"'. There is a 'Change Queue' button next to the Queue Name field.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Créer des planifications basées sur les événements avec Enterprise Manager

La page Create Schedule vous permet de choisir entre une planification standard, une planification basée sur une date/heure et une planification basée sur les événements. Si vous optez pour une planification basée sur les événements, l'interface change. Vous pouvez alors indiquer le nom de la file d'attente, le nom de l'agent et la condition associée à l'événement, en plus des autres attributs de la planification.

Remarque : Le planificateur exécute le travail chaque fois qu'il se produit un événement correspondant à `event_condition`. Toutefois, si l'événement intervient alors que le travail est déjà en cours d'exécution, il est ignoré. Il est consommé, mais il ne déclenche pas de nouvelle exécution du travail.

Références :

- Reportez-vous au manuel *Oracle Database Backup and Recovery Advanced User's Guide* pour plus d'informations sur la façon de créer des files d'attente et d'y placer des messages.
- Pour plus d'informations sur les règles Oracle Streams AQ et les conditions d'événement, reportez-vous à la description de la procédure `DBMS_AQADM.ADD_SUBSCRIBER` dans le manuel *Oracle Database PL/SQL Packages and Types Reference 11*.

Créer un travail basé sur les événements

Exemple : Créer un travail qui s'exécute si un fichier de données à charger en mode batch arrive dans le système de fichiers avant 09:00.

```
BEGIN
  DBMS_SCHEDULER.CREATE_JOB (
    job_name=>'ADMIN.PERFORM_DATA_LOAD',
    job_type => 'EXECUTABLE',
    job_action => '/loadaddir/start_my_load.sh',
    start_date => SYSTIMESTAMP,
    event_condition => 'tab.user_data.object_owner =
event_condition => 'tab.user_data.object_owner =
  'HR' and tab.user_data.object_name = 'DATA.TXT'
  and tab.user_data.event_type = 'FILE_ARRIVAL'
  and tab.user_data.event_timestamp < 9 ',
    queue_spec => 'HR.LOAD_JOB_EVENT_Q');
```

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Créer un travail basé sur les événements

Pour définir les informations relatives à des événements en tant qu'attributs d'un travail, vous pouvez utiliser une syntaxe de la procédure `CREATE_JOB` qui inclut les attributs `queue_spec` et `event_condition`. Les données d'événement peuvent être incluses directement dans le code définissant le travail, ou bien vous pouvez indiquer ces données en pointant sur une planification basée sur les événements. L'exemple de la diapositive ci-dessus utilise une planification basée sur un événement inclus dans le code du travail.

Il définit un travail qui est lancé lorsqu'un fichier arrive au niveau du système d'exploitation, à condition que cette arrivée se produise avant 09:00. Supposons que les données du message soient un objet comprenant quatre attributs nommés `object_owner`, `object_name`, `event_type` et `event_timestamp`.

L'exemple emploie un événement défini par l'utilisateur. Par conséquent, pour que le travail puisse être lancé lorsque le fichier arrive dans le système de fichiers, il faut qu'un programme ou une procédure place dans la file d'attente des événements indiquée le type d'objet généré par l'événement comprenant les informations appropriées. La file d'attente `HR.LOAD_JOB_EVENT_Q` doit être du même type que l'objet utilisé pour avertir le planificateur qu'un événement s'est produit. En d'autres termes, la file d'attente `HR.LOAD_JOB_EVENT_Q` doit être une file d'attente typée dont le type comprend quatre attributs nommés `object_owner`, `object_name`, `event_type` et `event_timestamp`.

Pour plus d'informations sur la façon de créer des files d'attente et d'y placer des messages, reportez-vous au manuel *Oracle Streams Advanced Queuing User's Guide and Reference*.

Planification basée sur les événements

Types d'événement :

- Événements générés par l'utilisateur ou par une application
- Événements générés par le planificateur

Événements générés par les travaux du planificateur :

- JOB_STARTED
- JOB_SUCCEEDED
- JOB_FAILED
- JOB_BROKEN
- JOB_COMPLETED
- JOB_STOPPED
- JOB_SCH_LIM_REACHED
- JOB_DISABLED
- JOB_CHAIN_STALLED
- JOB_ALL_EVENTS
- JOB_RUN_COMPLETED
- JOB_OVER_MAX_DUR

Exemple d'événement généré :

```
DBMS_SCHEDULER.SET_ATTRIBUTE ('hr.do_backup',  
'raise_events', DBMS_SCHEDULER.JOB_FAILED);
```

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Planification basée sur les événements

Vous pouvez créer un travail qui référence directement un événement servant à déclencher son exécution au lieu de lui associer une planification. Il existe deux types d'événement :

- **Événements générés par l'utilisateur ou par une application :** L'événement devant être consommé par le planificateur peut être généré par une application. Le planificateur réagit en lançant un travail. Exemples : L'exécution d'un travail se termine. Un fichier arrive dans le système de fichiers. Un compte est verrouillé dans la base de données. Le stock atteint un seuil minimal.
- **Événements générés par le planificateur :** Le planificateur peut générer un événement pour indiquer un changement de statut qui se produit dans son environnement. Par exemple, le planificateur peut générer un événement lorsqu'un travail commence, se termine ou dépasse le temps d'exécution qui lui est imparti. Le consommateur de l'événement est une application qui exécute une action en réponse à cet événement.

Vous pouvez configurer un travail de sorte que le planificateur génère un événement lorsqu'il change de statut. Pour cela, définissez l'attribut de travail `raise_events`. Par défaut, un travail ne génère aucun événement de modification de statut tant que vous n'avez pas modifié l'attribut `raise_events` associé. Pour modifier cet attribut, vous devez d'abord créer le travail à l'aide de la procédure `CREATE_JOB`. Utilisez ensuite la procédure `SET_ATTRIBUTE` pour modifier la valeur par défaut de l'attribut. Dans l'exemple de la diapositive, le travail `hr.do_backup` est modifié afin qu'un événement soit généré en cas d'échec de son exécution.

Planification basée sur les événements (suite)

Dès lors que vous avez activé les événements liés à des modifications de statut pour un travail, le planificateur génère ces événements en plaçant des messages dans la file d'attente des événements par défaut, `SYS.SCHEDULER$_EVENT_QUEUE`.

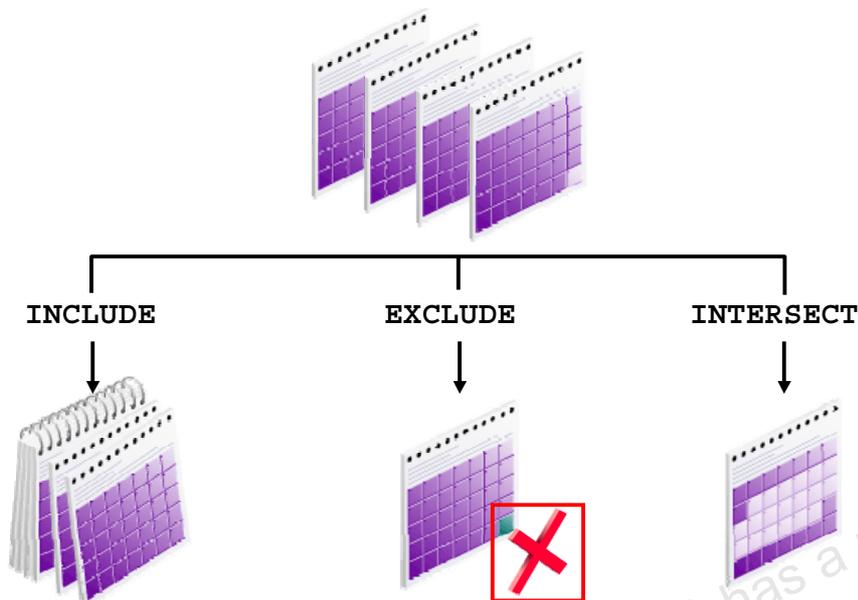
La file d'attente par défaut des événements du planificateur est sécurisée.

Selon l'application, vous pouvez être amené à la configurer de manière à ce que certains utilisateurs puissent s'en servir. Pour plus d'informations sur les files d'attente sécurisées, reportez-vous au manuel *Oracle Streams Concepts and Administration*.

La file d'attente par défaut des événements du planificateur est avant tout destinée aux événements générés par le planificateur. Oracle déconseille de l'employer pour les applications utilisateur et les événements définis par l'utilisateur.

Type d'événement	Description
JOB_STARTED	Le travail est lancé.
JOB_SUCCEEDED	L'exécution du travail s'est terminée avec succès.
JOB_FAILED	Le travail a échoué, soit en générant une erreur, soit en se terminant de façon anormale.
JOB_BROKEN	Le travail est désactivé et prend le statut BROKEN car le nombre d'échecs défini par l'attribut de travail MAX_FAILURES est dépassé.
JOB_COMPLETED	Le travail est terminé car il a atteint la valeur définie par l'attribut de travail MAX_RUNS ou END_DATE.
JOB_STOPPED	Le travail est arrêté par un appel de la procédure STOP_JOB.
JOB_SCH_LIM_REACHED	La limite définie par la planification du travail est atteinte. Le travail n'est pas lancé car le délai de lancement du travail a dépassé la valeur de l'attribut SCHEDULE_LIMIT.
JOB_DISABLED	Le travail est désactivé par le planificateur ou par un appel de la procédure SET_ATTRIBUTE.
JOB_CHAIN_STALLED	Un travail exécutant une chaîne prend le statut CHAIN_STALLED. La chaîne exécutée est bloquée si aucune étape n'est en cours d'exécution ou planifiée pour être exécutée et que la valeur NULL est attribuée à la chaîne EVALUATION_INTERVAL. La chaîne attend une intervention manuelle.
JOB_ALL_EVENTS	JOB_ALL_EVENTS n'est pas un événement mais une constante qui permet d'activer facilement tous les événements.
JOB_OVER_MAX_DUR	La durée d'exécution du travail a dépassé la limite autorisée.
JOB_RUN_COMPLETED	L'exécution d'un travail est terminée. Celui-ci peut avoir échoué, avoir réussi ou être arrêté.

Créer une planification complexe



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Créer une planification complexe

Une planification est un objet de base de données. Lorsque vous créez une planification, elle est automatiquement enregistrée. Vous pouvez créer une planification complexe par la combinaison de plusieurs autres. Cela vous permet d'ajouter ou d'exclure des dates spécifiques dans les résultats d'une expression de planification.

Vous pouvez utiliser les options suivantes pour définir l'intervalle de répétition d'une planification :

- **INCLUDE** : ajoute une liste de dates aux résultats de l'expression de planification.
- **EXCLUDE** : supprime une liste de dates des résultats de l'expression de planification.
- **INTERSECT** : utilise uniquement les dates qui sont communes à plusieurs planifications.

Lorsque vous créez des planifications devant être combinées, vous pouvez définir la liste des dates en incluant des dates codées en dur au format [YYYY] MMDD ou en incluant des planifications nommées créées avec la procédure CREATE_SCHEDULE. Par exemple, vous pouvez indiquer une liste de dates pour l'intervalle de répétition d'une planification en utilisant les valeurs suivantes :

```
0115,0315,0325,0615,quarter_end_dates,1215
```

Cette chaîne représente les dates suivantes : 15 janvier, 15 mars, 25 mars, 15 juin, 15 décembre, et toutes les dates définies par la planification QUARTER_END_DATES.

Lorsque vous n'indiquez pas l'année (facultative) dans les dates codées en dur d'une planification, ces dates sont incluses pour chaque année.

Quiz

Sélectionnez les affirmations qui sont vraies concernant les travaux légers persistants :

1. Ils occupent peu d'espace sur le disque pour les métadonnées de travail et pour le stockage des données d'exécution.
2. Ils offrent une flexibilité maximale.
3. Ils sont créés à partir d'un modèle de travail.
4. Ils peuvent être créés dans Enterprise Manager ou à partir de la ligne de commande.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Réponses : 1, 3

Laurent PERPETUE (laurent.perpetue@sr.fr) has a non-transferable license to use this Student Guide

Utiliser les notifications par e-mail

- Notifications par e-mail lors du changement de statut d'un travail
- Déclenchées par des événements de changement de statut
- Plusieurs notifications, plusieurs destinataires
- Vues *_SCHEDULER_NOTIFICATIONS

Utiliser les notifications par e-mail du planificateur

1. Indiquez l'adresse du serveur SMTP à utiliser pour l'envoi des messages :

```
DBMS_SCHEDULER.SET_SCHEDULER_ATTRIBUTE  
( 'email_server', 'host[:port]' );
```

2. De manière facultative, définissez l'adresse e-mail par défaut de l'expéditeur :

```
DBMS_SCHEDULER.SET_SCHEDULER_ATTRIBUTE  
( 'email_sender', 'valid_email_address' );
```

3. Associez des notifications par e-mail à un travail spécifique. (suite)

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Utiliser les notifications par e-mail

La fonctionnalité de notification par e-mail vous permet d'ajouter des notifications à des travaux existants pour que des événements particuliers associés au travail soient signalés par e-mail aux adresses indiquées. Pour chaque travail, vous pouvez ajouter des notifications pour plusieurs événements. Vous pouvez également envoyer la notification à plusieurs destinataires.

Pour activer la notification par e-mail, procédez comme suit :

1. Définissez l'attribut du planificateur `email_server`.
2. Utilisez l'attribut `email_sender` du planificateur pour définir l'adresse e-mail par défaut de l'expéditeur (facultatif).
3. Une fois le travail créé, exécutez la procédure `DBMS_SCHEDULER.ADD_JOB_EMAIL_NOTIFICATION` pour associer une ou plusieurs notifications au travail.

Le dictionnaire de données prend en charge les notifications par e-mail par le biais des vues *_SCHEDULER_NOTIFICATIONS.

Ajouter et supprimer des notifications par e-mail

```
DBMS_SCHEDULER.ADD_JOB_EMAIL_NOTIFICATION (
  job_name          IN VARCHAR2,
  recipients        IN VARCHAR2,
  sender            IN VARCHAR2 DEFAULT NULL,
  subject           IN VARCHAR2
  DEFAULT dbms_scheduler.default_notification_subject,
  body              IN VARCHAR2
  DEFAULT dbms_scheduler.default_notification_body,
  events            IN VARCHAR2
  DEFAULT 'JOB_FAILED, JOB_BROKEN, JOB_CHAIN_STALLED, JOB_OVER_MAX_DUR',
  filter_condition  IN VARCHAR2 DEFAULT NULL);
```

```
DBMS_SCHEDULER.REMOVE_JOB_EMAIL_NOTIFICATION (
  job_name          IN VARCHAR2,
  recipients        IN VARCHAR2 DEFAULT NULL,
  events            IN VARCHAR2 DEFAULT NULL);
```

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Ajouter et supprimer des notifications par e-mail

Pour associer des notifications par e-mail à un travail, utilisez la procédure `DBMS_SCHEDULER.ADD_JOB_EMAIL_NOTIFICATION`. Des e-mails sont envoyés aux adresses indiquées lorsque le travail génère l'un des événements répertoriés. Le travail est automatiquement modifié pour déclencher ces événements. Si une condition de filtrage est définie, seuls les événements répondant aux spécifications de l'argument de procédure `FILTER_CONDITION` génèrent un e-mail.

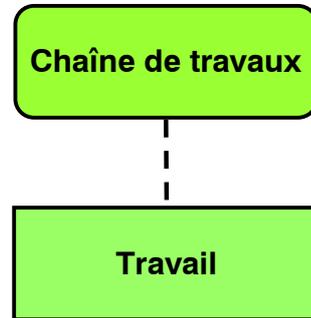
Cette procédure échoue si l'attribut du planificateur `EMAIL_SERVER` n'est pas défini ou si le travail défini n'existe pas. L'utilisateur qui appelle cette procédure doit être le propriétaire du travail, disposer du privilège système `CREATE ANY JOB` ou avoir reçu le privilège `ALTER` pour ce travail.

- L'élément **subject** des e-mails peut contenir les variables suivantes : `%job_owner%`, `%job_name%`, `%event_type%`, `%event_timestamp%`, `%log_id%`, `%error_code%`, `%error_message%`, `%run_count%`, `%failure_count%`, `%retry_count%`, `%job_subname%`, `%job_class_name%`.
- L'élément **body** des e-mails peut contenir les mêmes variables.
- La liste associée à l'élément **events** ne peut pas être `NULL`. Elle utilise la virgule comme séparateur. Reportez-vous à la liste des événements associés à l'attribut `RAISE_EVENTS` de la table `JOBS` pour connaître les événements valides.
- Si **filter_condition** est `NULL` (valeur par défaut), toutes les occurrences des événements indiqués donnent lieu à l'envoi d'e-mails aux destinataires précisés.

Pour supprimer des notifications par e-mail pour un travail, utilisez la procédure `DBMS_SCHEDULER.REMOVE_JOB_EMAIL_NOTIFICATION`.

Créer des chaînes de travaux

1. Créez un objet de type chaîne.
2. Définissez les étapes de la chaîne.
3. Définissez les règles de la chaîne.
4. Lancez la chaîne :
 - Activez la chaîne.
 - Créez un travail qui pointe sur la chaîne.



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Créer des chaînes de travaux

Une chaîne est une série nommée de programmes qui sont associés pour répondre à un objectif particulier. On parle de "planification de travaux interdépendants". Exemple : *Exécuter le programme A puis le programme B, mais n'exécuter le programme C que si les programmes A et B réussissent. Sinon, exécuter le programme D.*

Chaque position d'une chaîne de programmes interdépendants est appelée étape.

En général, une fois qu'un ensemble initial d'étapes de la chaîne a démarré, l'exécution des étapes successives dépend de la fin d'une ou de plusieurs étapes précédentes. Pour créer une chaîne et l'utiliser, effectuez les opérations ci-après, dans l'ordre indiqué. Toutes les procédures citées font partie du package DBMS_SCHEDULER, sauf indication contraire.

1. **Créez une chaîne** à l'aide de la procédure CREATE_CHAIN. Le nom de la chaîne peut être qualifié avec un nom de schéma (par exemple, *myschema.mynome*).
2. **Définissez les étapes de la chaîne.** Lorsque vous définissez une étape, vous lui attribuez un nom et vous indiquez ce qui se passe pendant cette étape. Chaque étape peut pointer vers :
 - un programme,
 - une autre chaîne (chaîne imbriquée),
 - un événement.

Vous définissez une étape qui pointe sur un programme ou sur une chaîne imbriquée en appelant la procédure DEFINE_CHAIN_STEP.

Créer des chaînes de travaux (suite)

Pour définir une étape qui attend qu'un événement se produise, utilisez la procédure `DEFINE_CHAIN_EVENT_STEP`. Les arguments de la procédure peuvent pointer vers une planification basée sur les événements ou peuvent inclure directement dans le code une spécification de file d'attente et une condition d'événement. Une étape qui pointe sur un événement attend que cet événement soit généré. Si l'événement se produit, l'étape réussit.

3. Après avoir créé l'objet chaîne, **définissez les règles de la chaîne**. Les règles définissent à quel moment les étapes sont exécutées et indiquent les dépendances entre les étapes. Chaque règle comporte une *condition* et une *action* :

- Si la condition prend la valeur `TRUE`, l'action est exécutée. La condition peut contenir toute syntaxe qui est valide dans une clause `SQL WHERE`. Les conditions sont généralement basées sur le résultat d'une ou de plusieurs étapes précédentes. Par exemple, vous pouvez décider d'exécuter une étape si les deux étapes précédentes ont réussi, et d'en exécuter une autre si l'une des deux étapes précédentes a échoué.
- L'action indique ce qui doit être fait lorsque la condition est remplie. Une action classique consiste à exécuter une étape précise. Une action peut consister à lancer ou à arrêter une étape. Vous pouvez également choisir d'interrompre l'exécution de la chaîne de travaux et de renvoyer soit une valeur, soit un nom d'étape et un code d'erreur.

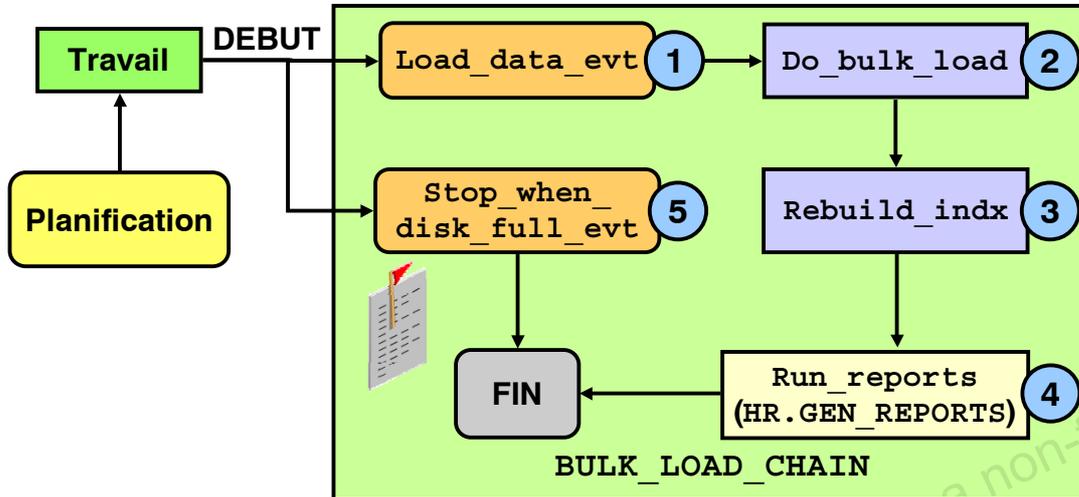
Toutes les règles ajoutées à une chaîne sont utilisées conjointement pour définir le comportement global de la chaîne. Une fois le travail commencé, les règles sont toutes évaluées à la fin de chaque étape afin de déterminer les actions suivantes. Pour ajouter une règle à une chaîne, utilisez la procédure `DEFINE_CHAIN_RULE`. Appelez cette procédure une fois pour chaque règle que vous souhaitez ajouter à la chaîne.

4. **Lancez la chaîne**. Cette opération implique deux actions :

- Activez la chaîne avec la procédure `ENABLE`. (Lorsqu'elle est créée, une chaîne est toujours désactivée. Vous pouvez ainsi lui associer des étapes et des règles avant son exécution par un travail.) Vous ne recevez pas de message d'erreur si vous activez une chaîne qui est déjà active.
- Pour exécuter la chaîne, vous devez créer un travail de type `'CHAIN'`. L'action associée au travail doit faire référence au nom de la chaîne. Pour ce travail, vous pouvez utiliser des planifications basées sur les événements ou sur une date/heure.

Exemple de chaîne

Planification de travaux interdépendants



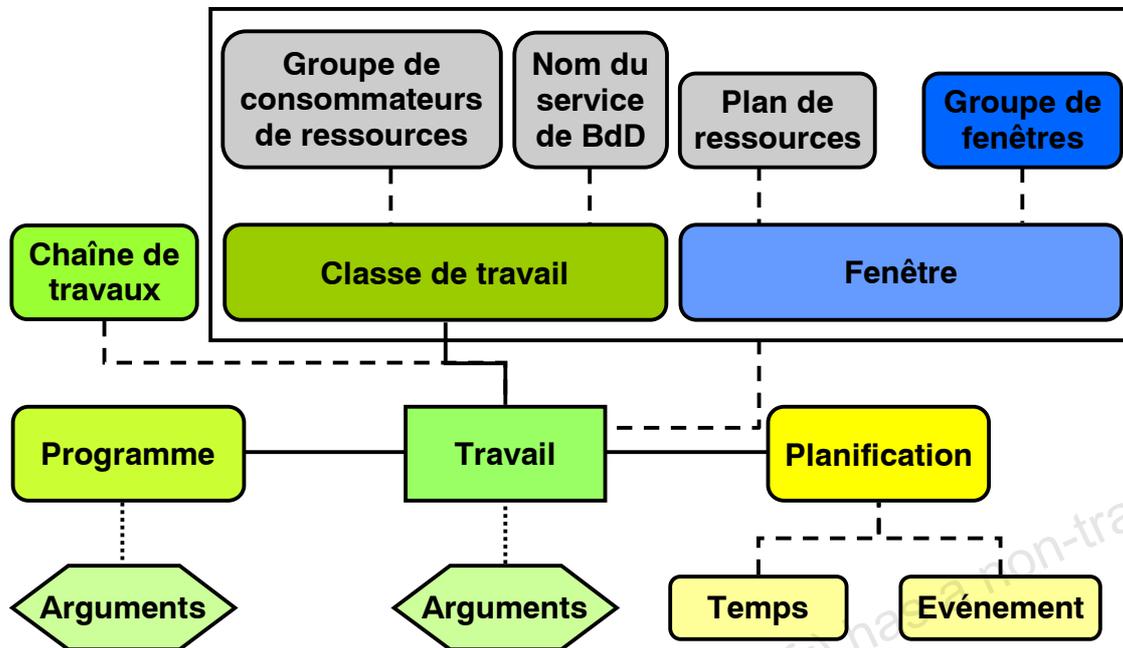
ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Exemple de chaîne

L'ensemble des tâches et conditions qui se produisent pendant un chargement en masse de données constitue un exemple de chaîne. Vous devez d'abord disposer de données à charger. Vous chargez ensuite les données, en vérifiant au préalable que l'espace disponible dans le système de fichiers est suffisant pour cette opération. Une fois les données chargées, vous devez reconstruire les index définis sur les tables mises à jour. Vous pouvez ensuite exécuter des états sur les données nouvellement chargées. La diapositive ci-dessus illustre un exemple de planification de travaux interdépendants.

Fonctionnalités avancées du planificateur



Copyright © 2009, Oracle. Tous droits réservés.

ORACLE

Fonctionnalités avancées du planificateur

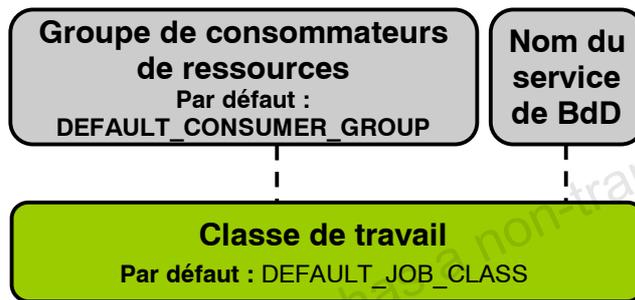
Les fonctionnalités avancées du planificateur permettent un contrôle plus précis de certains aspects de la planification tels que les fenêtres de travail et la hiérarchisation des travaux par priorité. Ces fonctionnalités avancées sont présentées ci-après, puis étudiées en détail dans les diapositives qui suivent.

- Une **fenêtre** est un intervalle avec un début et une fin bien définis qui est utilisé pour activer différents plans de ressources à différents instants. Cela vous permet de modifier l'allocation des ressources au cours d'une période, par exemple une journée ou une année.
- Un **groupe de fenêtres** représente une liste de fenêtres et facilite la gestion des fenêtres. Vous pouvez utiliser une fenêtre ou un groupe de fenêtres pour la planification d'un travail afin de garantir que le travail s'exécute uniquement lorsqu'une fenêtre et le plan de ressources associé sont actifs.
- Une **classe de travail** définit une catégorie de travaux qui partagent des exigences communes en matière d'utilisation des ressources, ainsi que d'autres caractéristiques. Elle sert à regrouper des travaux.
- Le **groupe de consommateurs de ressources** associé à une classe de travail détermine les ressources allouées aux travaux de cette classe.
- Un **plan de ressources** permet aux utilisateurs d'affecter des priorités aux ressources (essentiellement la CPU) parmi les différents groupes de consommateurs de ressources.

Remarque : Les objets sur fond gris ne sont pas des objets du planificateur.

Classes de travaux

- Elles permettent d'affecter le même ensemble de valeurs d'attribut à des travaux membres.
- Elles sont créées à l'aide de la procédure `CREATE_JOB_CLASS`.
- Vous associez des travaux à une classe à l'aide de la procédure `SET_ATTRIBUTE`.
- Les classes appartiennent au schéma `SYS`.
- Elles définissent l'allocation des ressources pour les travaux membres.
- Affectez à l'attribut de service le nom d'un service de base de données.
- Les classes permettent de regrouper des travaux pour leur affecter des priorités.



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Classes de travaux

Une classe de travail est une stratégie pour les travaux qu'elle regroupe. Elle définit différents attributs tels que le niveau de journalisation. Lorsque vous affectez un travail à une classe, il hérite de ces attributs. Par exemple, vous pouvez indiquer la même stratégie pour purger les entrées de journalisation de tous les travaux de paie.

- Vous pouvez utiliser la procédure `CREATE_JOB_CLASS` pour créer une classe de travail. Une classe appartient toujours au schéma `sys`. Pour créer une classe, vous devez disposer du privilège `MANAGE_SCHEDULER`. Il existe une classe de travail par défaut nommée `DEFAULT_JOB_CLASS`, créée avec la base de données.
- Après avoir créé une classe de travail, vous pouvez lui affecter des travaux, lors de la création de ceux-ci, ou postérieurement à l'aide de la procédure `SET_ATTRIBUTE` du package `DBMS_SCHEDULER`. Si un travail n'est associé à aucune classe, il appartient à cette classe par défaut.
- Affectez à l'attribut de service d'une classe de travail le nom de service de base de données de votre choix. Cela détermine les instances d'un environnement Real Application Clusters qui exécutent les travaux membres et, éventuellement, les ressources système affectées aux travaux membres.

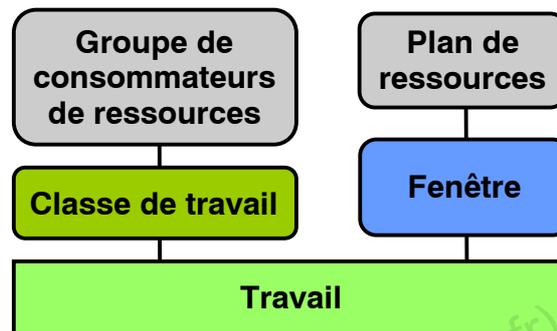
Classes de travaux (suite)

- Les classes de travaux définissent l'allocation des ressources pour les travaux membres. Elles permettent de relier Database Resource Manager et le planificateur, car chacune d'elles peut indiquer un groupe de consommateurs de ressources comme attribut. Les travaux membres appartiennent alors au groupe de consommateurs de ressources indiqué et se voient affecter des ressources en fonction des paramètres du plan de ressources en cours. Vous pouvez aussi conserver la valeur NULL de l'attribut `resource_consumer_group` et affecter à l'attribut de service d'une classe le nom de service de base de données de votre choix. Ce service peut à son tour être mis en correspondance avec un groupe de consommateurs de ressources. Si l'attribut `resource_consumer_group` et l'attribut de service sont définis, et si le service désigné est mis en correspondance avec un groupe de consommateurs de ressources, le groupe de consommateurs de ressources nommé dans l'attribut `resource_consumer_group` est prioritaire. Si aucun groupe de consommateurs de ressources n'est désigné lors de la création d'une classe de travail, cette dernière est mise en correspondance avec le groupe de consommateurs de ressources `DEFAULT_CONSUMER_GROUP`. Il est possible que les ressources allouées aux travaux de la classe de travail par défaut ou d'une classe de travail associée au groupe de consommateurs de ressources par défaut ne soient pas suffisantes pour leur permettre d'effectuer leurs tâches lorsque Resource Manager est activé.
- Les classes permettent de regrouper des travaux pour leur affecter des priorités. Au sein de la même classe, vous pouvez affecter des valeurs de priorité de 1 à 5 aux différents travaux. Si deux travaux sont planifiés pour démarrer à la même heure, celui qui a la valeur la plus élevée est prioritaire. Ainsi, un travail de moindre importance ne risque pas d'entraver l'exécution en temps utile d'un travail critique. Si deux travaux ont la même priorité, celui qui présente la date de début la plus proche est prioritaire. Si aucune priorité n'est affectée à un travail, la priorité par défaut de 3 lui est attribuée.

Fenêtres

Les fenêtres du planificateur présentent les caractéristiques suivantes :

- Elles permettent de démarrer des travaux ou de modifier l'allocation des ressources aux travaux pour différentes périodes.
- Une seule fenêtre est active à un instant donné.
- Les fenêtres sont créées à l'aide de la procédure `CREATE_WINDOW`.



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Fenêtres

La priorité des travaux peut évoluer dans le temps. Par exemple, pendant la nuit, vous pouvez allouer un pourcentage élevé des ressources aux travaux de chargement de data warehouse et, pendant la journée, affecter la majeure partie des ressources aux travaux applicatifs. Pour ce faire, vous pouvez modifier le plan de ressources de la base de données dans une fenêtre du planificateur.

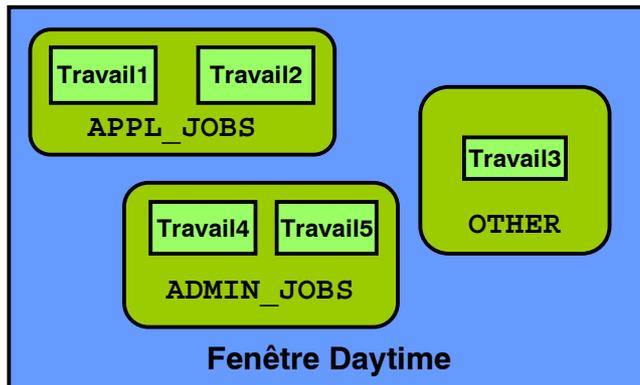
- Les fenêtres du planificateur permettent de démarrer des travaux automatiquement ou de modifier l'allocation des ressources aux travaux pour différentes périodes du jour, de la semaine, etc. Une fenêtre est représentée par un intervalle de temps avec un début et une fin bien définis ("de minuit à 6:00", par exemple).
- Une seule fenêtre peut être active à un instant donné.
- Vous pouvez créer une fenêtre à l'aide de la procédure `CREATE_WINDOW`.

Associées aux classes de travaux, les fenêtres permettent de contrôler l'allocation des ressources. Chaque fenêtre indique le plan d'allocation des ressources à activer à son ouverture (lorsqu'elle devient active), et chaque classe indique un groupe de consommateurs de ressources ou un service de base de données, qui peut être mis en correspondance avec un groupe de consommateurs de ressources. Un travail qui s'exécute dans une fenêtre se voit donc allouer des ressources en fonction du groupe de consommateurs de ressources de sa classe de travail et du plan de ressources de la fenêtre (comme indiqué dans la diapositive).

Affecter des priorités aux travaux dans une fenêtre

Affecter des priorités aux travaux :

- Au niveau classe (via des plans de ressources)
- Au niveau travail (via un attribut de priorité)
- L'affectation des priorités n'est pas garantie pour les travaux de différentes classes.



Travail	Priorité
Travail1	1
Travail2	2
Travail3	3
Travail4	5
Travail5	2

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Affecter des priorités aux travaux dans une fenêtre

Lors de la création de plusieurs travaux dans une base de données, vous devez disposer d'un moyen d'adapter le traitement des travaux à vos besoins métier, ainsi que d'indiquer les travaux qui sont prioritaires. Pour une fenêtre particulière, plusieurs classes de travaux peuvent être en cours d'exécution, chacune avec sa propre priorité.

Il existe deux niveaux d'affectation de priorités aux travaux : le niveau classe et le niveau travail.

- La première affectation de priorité se situe au niveau classe, par l'intermédiaire de plans de ressources. L'affectation de priorités aux travaux des différentes classes est effectuée uniquement sur la base de l'allocation des ressources aux classes.
- La deuxième affectation de priorité a lieu dans la classe, avec l'attribut de priorité du travail.

Les niveaux de priorité ne sont importants que lorsque deux travaux de la même classe sont supposés démarrer en même temps. Le travail avec la priorité la plus élevée démarre en premier.

L'affectation des priorités n'est pas garantie pour les travaux de différentes classes.

Par exemple, il se peut qu'un travail de priorité élevée dans la classe `APPL_JOBS` ne démarre pas avant un travail de faible priorité dans la classe `ADMIN_JOBS`, même s'ils partagent la même planification. Si la classe de travail `APPL_JOBS` dispose d'un niveau de ressources inférieur, le travail de priorité élevée de cette classe doit attendre que les ressources soient disponibles, même si des ressources sont disponibles pour des travaux de priorité inférieure dans une classe de travail différente.

Créer un ensemble de travaux

1. Déclarez des variables de type `sys.job` et `sys.job_array` :

```
DECLARE
  newjob sys.job;
  newjobarr sys.job_array;
```

2. Initialisez l'ensemble de travaux :

```
BEGIN
  newjobarr := SYS.JOB_ARRAY();
```

3. Dimensionnez l'ensemble de travaux de sorte qu'il contienne le nombre de travaux nécessaires :

```
newjobarr.EXTEND(100);
```

(... suite)

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Créer un ensemble de travaux

Une méthode plus efficace pour regrouper des travaux consiste à utiliser un ensemble de travaux (job array). Cela s'applique également aux travaux légers. Dans l'exemple de la diapositive, 100 spécifications de travail sont créées dans un ensemble de travaux et soumises à la file d'attente des travaux en une transaction unique. Notez que pour un travail léger, une quantité très limitée d'informations est nécessaire. Dans l'exemple, le paramètre `start_time` prend la valeur par défaut NULL. Le travail est donc planifié pour démarrer immédiatement.

1. Déclarez une variable pour la définition de travail et une autre pour l'ensemble de travaux.
2. Initialisez l'ensemble de travaux à l'aide du constructeur `SYS.JOB_ARRAY`. Une place est ainsi créée dans l'ensemble.
3. Définissez la taille de l'ensemble en fonction du nombre de travaux attendus.
4. Créez chaque travail et placez-le dans l'ensemble. Dans l'exemple de la diapositive, la seule différence porte sur le nom du travail. La variable `start_time` du travail est omise et prend donc la valeur par défaut NULL, ce qui indique que le travail s'exécutera immédiatement.
5. Utilisez la procédure `CREATE_JOBS` pour soumettre tous les travaux de l'ensemble en une seule transaction.

Remarque : Si l'ensemble de travaux est de très petite taille, les performances ne sont pas notablement meilleures qu'avec des travaux uniques.

Créer un ensemble de travaux

4. Placez des travaux dans l'ensemble :

```
FOR i IN 1..100 LOOP
    newjob := SYS.JOB(job_name => 'LWTJK' || to_char(i),
                     job_style => 'LIGHTWEIGHT',
                     job_template => 'MY_PROG',
                     enabled => TRUE );
    newjobarr(i) := newjob;
END LOOP;
```

5. Soumettez l'ensemble de travaux en une seule transaction :

```
DBMS_SCHEDULER.CREATE_JOBS(newjobarr,
                            'TRANSACTIONAL');
```

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Créer un ensemble de travaux (suite)

Voici le code complet de l'exemple :

```
DECLARE
    newjob sys.job;
    newjobarr sys.job_array;
BEGIN
    -- Create an array of JOB object types
    newjobarr := sys.job_array();
    -- Allocate sufficient space in the array
    newjobarr.extend(100);
    -- Add definitions for jobs
    FOR i IN 1..100 LOOP
        -- Create a JOB object type
        newjob := sys.job(job_name => 'LWTJK' || to_char(i),
                         job_style => 'LIGHTWEIGHT',
                         job_template => 'PROG_1',
                         enabled => TRUE );

        -- Add job to the array
        newjobarr(i) := newjob;
    END LOOP;
    -- Call CREATE_JOBS to create jobs in one transaction
    DBMS_SCHEDULER.CREATE_JOBS(newjobarr, 'TRANSACTIONAL');
END;
```

/

Quiz

Sélectionnez toutes les affirmations qui sont vraies à propos du planificateur et de ses fonctionnalités :

1. Les travaux légers peuvent être créés au moyen d'un ensemble de travaux.
2. Les priorités de travaux au niveau classe (via les plan de ressources) et au niveau travail (via l'attribut de priorité) sont mutuellement exclusives.
3. Associées aux classes de travaux, les fenêtres permettent de contrôler l'allocation des ressources.
4. Les chaînes de travaux permettent d'implémenter la "planification de travaux interdépendants".

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Réponses : 1, 3, 4

Créer un contrôleur de fichier et un travail fondé sur les événements

Effectuez les opérations suivantes :

1. Dans le planificateur, créez un objet de type informations d'identification et de connexion et associez-lui le privilège EXECUTE.
2. Créez un contrôleur de fichier et associez-lui le privilège EXECUTE.
3. Dans le planificateur, créez un objet de type programme avec un argument de métadonnées référant le message d'événement.
4. Créez un travail fondé sur les événements référant le contrôleur de fichier. Activez le travail pour exécuter chaque instance de l'événement relatif à l'arrivée du fichier (facultatif).
5. Activez le contrôleur de fichier, le programme et le travail.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Créer un contrôleur de fichier et un travail fondé sur les événements

Pour créer un contrôleur de fichier et le travail fondé sur les événements qui démarre à l'arrivée du fichier désigné, effectuez les opérations suivantes :

1. Dans le planificateur, créez un objet de type informations d'identification et de connexion à utiliser pour l'authentification auprès du système d'exploitation hôte en vue d'accéder aux fichiers. Accordez le privilège EXECUTE sur cet objet au schéma qui possède le travail fondé sur les événements que le contrôleur de fichier va démarrer.
2. Créez un contrôleur de fichier. Accordez le privilège EXECUTE sur ce contrôleur à tout schéma qui possède un travail fondé sur les événements référant ce contrôleur.
3. Dans le planificateur, créez un objet de type programme avec un argument de métadonnées référant le message d'événement.
 - Définissez l'argument de métadonnées à l'aide de l'attribut EVENT_MESSAGE.
 - Créez la procédure stockée à l'aide d'un argument de type SYS.SCHEDULER_FILEWATCHER_RESULT appelé par le programme. Celle-ci doit être associée à un argument de type SYS.SCHEDULER_FILEWATCHER_RESULT, qui représente le type de données du message d'événement. La position de cet argument doit correspondre à celle de l'argument de métadonnées défini. La procédure peut accéder aux attributs de ce type de données abstrait pour en savoir plus sur le fichier arrivé.

Créer un contrôleur de fichier et un travail fondé sur les événements (suite)

4. Créez un travail fondé sur les événements référençant le contrôleur de fichier.
Vous pouvez utiliser la procédure `DBMS_SCHEDULER.SET_ATTRIBUTE` pour activer le travail à exécuter pour chaque instance de l'événement relatif à l'arrivée du fichier, même si le travail est déjà en train de traiter un événement précédent.
Définissez l'attribut `PARALLEL_INSTANCES` avec la valeur `TRUE`.

```
BEGIN
  DBMS_SCHEDULER.SET_ATTRIBUTE (' ', 'PARALLEL_INSTANCES',
  TRUE);
END;
```

Cette opération permet au travail de s'exécuter en tant que travail léger pour que plusieurs instances du travail puissent être démarrées rapidement.

Si `PARALLEL_INSTANCES` est défini avec la valeur `FALSE` (valeur par défaut), les événements du contrôleur de fichier qui se produisent alors que le travail fondé sur les événements est déjà en train de traiter un autre sont supprimés.

5. Activez le contrôleur de fichier, le programme et le travail.

Activer les événements relatifs à l'arrivée du fichier à partir de systèmes distants

Pour activer le déclenchement des événements relatifs à l'arrivée du fichier sur des systèmes distants, effectuez les opérations suivantes :

1. Paramétrez la base de données pour exécuter des travaux externes distants.
2. Installez, configurez, enregistrez et démarrez l'agent de planification sur le premier système distant.
3. Répétez l'étape 2 pour chaque système distant supplémentaire.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Activer les événements relatifs à l'arrivée du fichier à partir de systèmes distants

Pour recevoir les événements relatifs à l'arrivée du fichier à partir d'un système distant, vous devez installer l'agent de planification sur ce système, puis enregistrer l'agent dans la base de données. Le système distant ne nécessite pas d'instance Oracle Database pour générer de tels événements.

Pour plus d'informations, reportez-vous au manuel *Oracle Database Administrator's Guide 11g Release 2*.

Planifier des travaux sur une base de données distants

- Créez un travail qui exécute des procédures stockées et des blocs PL/SQL anonymes sur une autre instance de base de données, sur le même hôte ou sur un hôte distant.
- La version de la base de données cible peut être n'importe quelle version d'Oracle Database.
- `DBMS_SCHEDULER.CREATE_DATABASE_DESTINATION` et `DBMS_SCHEDULER.CREATE_CREDENTIAL` peuvent être utilisés pour les travaux de base de données distants.
- Les travaux de type `PLSQL_BLOCK` et `STORED_PROCEDURE` peuvent être le sujet d'appels `SET_ATTRIBUTE` pour les attributs `DESTINATION` et `CREDENTIAL`.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Planifier des travaux sur une base de données distants

Vous pouvez à présent créer un travail qui exécute des procédures stockées et des blocs PL/SQL anonymes sur une autre instance de base de données, sur le même hôte ou sur un hôte distant. La version de la base de données cible peut être n'importe quelle version d'Oracle Database.

Il n'existe aucune nouvelle procédure pour prendre en charge les travaux effectués sur une base de données distante, mais des modifications ont été apportées aux procédures `DBMS_SCHEDULER` existantes pour cette fonctionnalité. La page suivante présente des informations supplémentaires.

Créer des travaux sur une base de données distante

Pour créer un travail distant, effectuez les opérations suivantes :

1. Paramétrez la base de données d'origine pour les travaux distants.
2. Créez le travail à l'aide de `DBMS_SCHEDULER.CREATE_JOB`.
3. Créez les informations d'identification et de connexion à l'aide de `DBMS_SCHEDULER.CREATE_CREDENTIAL`.
4. Définissez l'attribut `CREDENTIAL_NAME` du travail en utilisant `DBMS_SCHEDULER.SET_ATTRIBUTE`.
5. Définissez l'attribut `DESTINATION` du travail en utilisant `DBMS_SCHEDULER.SET_ATTRIBUTE`.
6. Activez le travail à l'aide de `DBMS_SCHEDULER.ENABLE`.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Créer des travaux sur une base de données distante

Vous pouvez exécuter les tâches répertoriées ci-dessus pour créer un travail sur une base de données distante.

Pour paramétrer la base de données d'origine pour des travaux distants, effectuez les opérations suivantes :

1. Vérifiez qu'Oracle XML DB est bien installé.
2. Activez les connexions HTTP à la base de données.

```
BEGIN
DBMS_XDB.SETHTTPPORT(port);
END;
```
3. Exécutez le script `prvtrsch.plb`.
4. Définissez un mot de passe d'enregistrement pour les agents de planification :

```
BEGIN
DBMS_SCHEDULER.SET_AGENT_REGISTRATION_PASS('password');
END;
```

Pour obtenir un exemple détaillé, reportez-vous au manuel *Oracle Database Administrator's Guide 11g Release 2*.

Planifier des travaux à destinations multiples

- Cette fonctionnalité vous permet de définir plusieurs cibles sur lesquelles les travaux doivent s'exécuter.
- Elle fournit la possibilité de surveiller et de contrôler les travaux à partir de la base sur laquelle ils ont été créés.
- Lors de son exécution, un travail à destinations multiples est affiché en tant que collection de travaux qui sont des copies presque identiques.
- Tous les travaux s'exécutent selon le fuseau horaire indiqué dans la date de départ du travail ou utilisent le fuseau horaire de la base de données source.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Planifier des travaux à destinations multiples

Cette fonctionnalité vous permet de définir plusieurs cibles sur lesquelles les travaux doivent s'exécuter. Vous avez la possibilité de surveiller et de contrôler les travaux à partir de la base sur laquelle vous les avez créés. Vous pouvez aussi effectuer les opérations suivantes :

- Définir plusieurs bases de données ou machines sur lesquelles un travail doit s'exécuter.
- Modifier un travail planifié sur plusieurs cibles en tant qu'entité unique.
- Arrêter ou supprimer des travaux s'exécutant sur une ou plusieurs cibles distantes.
- Trouver le statut des instances de travail sur toutes les cibles d'un travail.

Remarque : Dans la version initiale de cette fonctionnalité, toutes les destinations s'exécutent selon le fuseau horaire indiqué dans la date de départ du travail ou utilisent par défaut le fuseau horaire de la base de données source.

Afficher les métadonnées du planificateur

Principales vues de gestion du planificateur :

- *_SCHEDULER_JOBS affiche tous les travaux, activés ou non.
- *_SCHEDULER_SCHEDULES affiche toutes les planifications.
- *_SCHEDULER_PROGRAMS affiche tous les programmes.
- *_SCHEDULER_RUNNING_JOBS affiche le statut des travaux actifs.
- *_SCHEDULER_JOBS affiche tous les changements de statut des travaux.
- *_SCHEDULER_JOB_RUN_DETAILS affiche toutes les exécutions de travaux terminées.

```
SELECT job_name, status, error#, run_duration
FROM USER_SCHEDULER_JOB_RUN_DETAILS;
```

JOB_NAME	STATUS	ERROR#	RUN_DURATION
GATHER_STATS_JOB	SUCCESS	0	+000 00:08:20
PART_EXCHANGE_JOB	FAILURE	6576	+000 00:00:00

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Afficher les métadonnées du planificateur

La table JOBS contient l'ensemble des travaux. Il en existe une par base de données. Cette table stocke des informations concernant tous les travaux, comme le nom du propriétaire ou le niveau de journalisation. Ces informations figurent dans les vues *_SCHEDULER_JOBS.

Les travaux étant des objets de base de données, ils peuvent s'accumuler et, au final, devenir encombrants. Afin d'y remédier, les objets de type JOB sont supprimés automatiquement par défaut après exécution. Ce comportement est contrôlé par l'attribut de travail auto_drop.

De nombreuses vues sont à la disposition du DBA et des utilisateurs dotés des privilèges appropriés. Elles fournissent des informations opérationnelles concernant le planificateur, les travaux, les planifications, les fenêtres, etc. Il s'agit des vues suivantes :

- *_SCHEDULER_PROGRAM_ARGS affiche tous les arguments définis pour tous les programmes, ainsi que les éventuelles valeurs par défaut.
- *_SCHEDULER_JOBS affiche tous les travaux, activés ou non.
- *_SCHEDULER_JOB_RUN_DETAILS affiche toutes les exécutions de travaux terminées (qu'elles aient réussi ou non). Elle comprend une ligne pour chaque instance de travail. Chaque ligne contient des informations sur l'exécution du travail de cette instance. ERROR# est le numéro de la première erreur rencontrée.
- *_SCHEDULER_GLOBAL_ATTRIBUTE affiche les valeurs actuelles des attributs du planificateur.
- *_SCHEDULER_JOB_ARGS affiche toutes les valeurs d'argument définies pour tous les travaux.
- *_SCHEDULER_JOB_CLASSES affiche toutes les classes de travaux.

Afficher les métadonnées du planificateur (suite)

Pour les chaînes de travaux :

- *_SCHEDULER_RUNNING_CHAINS affiche toutes les chaînes actives.
- *_SCHEDULER_CHAIN_STEPS affiche toutes les étapes de l'ensemble des chaînes.
- *_SCHEDULER_CHAINS affiche toutes les chaînes.
- *_SCHEDULER_CHAIN_RULES affiche toutes les règles de l'ensemble des chaînes.

Pour les fenêtres et les autres objets avancés :

- *_SCHEDULER_WINDOWS affiche toutes les fenêtres.
- *_SCHEDULER_WINDOW_GROUPS affiche tous les groupes de fenêtres.
- *_SCHEDULER_WINDOWGROUP_MEMBERS affiche les membres de tous les groupes de fenêtres (une ligne pour chaque membre de groupe).
- *_SCHEDULER_JOB_LOG affiche tous les changements d'état relatifs à des travaux.
- *_SCHEDULER_CREDENTIALS affiche la liste des informations d'identification et de connexion définies dans la base de données (les mots de passe sont brouillés).
- *_SCHEDULER_JOB_ROLES affiche l'ensemble des travaux par rôle de base de données.

Les travaux légers apparaissent dans les mêmes vues que les travaux standards :

- *_SCHEDULER_JOBS affiche tous les travaux, y compris ceux pour lesquels JOB_STYLE= 'LIGHTWEIGHT'.
- *_SCHEDULER_JOB_ARGS affiche toutes les valeurs d'argument définies, y compris pour les travaux légers.
- En revanche, puisque les travaux légers ne sont pas des objets de base de données, ils ne sont pas visibles par l'intermédiaire des vues *_OBJECTS.

Depuis Oracle Database 11gR2 :

- *_SCHEDULER_NOTIFICATIONS affiche les notifications par e-mail définies.
- *_SCHEDULER_FILE_WATCHERS affiche les informations de configuration relatives au contrôleur de fichier.

Les vues suivantes affichent des informations sur les travaux à destinations multiples :

- *_SCHEDULER_DESTS affiche toutes les destinations vers lesquelles des travaux distants peuvent être planifiés. Celles-ci incluent les destinations externes (pour les travaux externes distants) ainsi que les destinations de base de données pour les travaux distants.
- *_SCHEDULER_EXTERNAL_DESTS affiche tous les agents enregistrés auprès de la base de données qui peuvent être utilisés comme destination pour des travaux externes distants.
- *_SCHEDULER_DB_DESTS affiche toutes les bases de données sur lesquelles vous pouvez planifier des travaux de base de données distants.
- *_SCHEDULER_GROUPS affiche les groupes de votre schéma ou tous les groupes de la base de données.
- *_SCHEDULER_GROUP_MEMBERS affiche les membres du groupe de votre schéma ou tous les membres du groupe de la base de données.
- *_SCHEDULER_JOB_DESTS affiche le statut d'un travail effectué sur une base de données distante.

Remarque : Dans les vues répertoriées ci-dessus, l'astérisque figurant au début du nom d'une vue peut être remplacé par DBA, ALL ou USER.

Quiz

Sélectionnez les affirmations qui sont vraies concernant le planificateur Oracle :

1. La création de travaux sur une base de données distante est une tâche manuelle nécessitant l'utilisation de commandes du système d'exploitation.
2. Dans le planificateur, un objet de type informations d'identification et de connexion sert à l'authentification auprès du système d'exploitation hôte pour l'accès aux fichiers.
3. Vous pouvez indiquer plusieurs cibles sur lesquelles les travaux peuvent s'exécuter et les surveiller à partir de la base sur laquelle elles ont été créées.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Réponses : 2, 3

Synthèse

Ce chapitre vous a permis d'apprendre à :

- simplifier les tâches de gestion à l'aide du planificateur
- créer un travail, un programme et une planification
- surveiller l'exécution d'un travail
- utiliser une planification basée sur une date/heure ou sur les événements pour exécuter les travaux du planificateur
- décrire l'utilisation des fenêtres, des groupes de fenêtres, des classes de travaux et des groupes de consommateurs
- utiliser la notification par e-mail
- utiliser des chaînes de travaux pour effectuer une série de tâches interdépendantes
- décrire les travaux du planificateur sur les systèmes distants
- utiliser les fonctionnalités avancées du planificateur pour définir la priorité des travaux

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Présentation de l'exercice 17 : Automatiser des tâches avec le planificateur

Cet exercice porte sur les points suivants :

- Créer un travail qui exécute un programme en dehors de la base de données
- Créer un programme et une planification
- Créer un travail qui utilise un programme et une planification
- Créer un travail léger
- Surveiller l'exécution des travaux

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Remarque

Cet exercice utilise à la fois Enterprise Manager Database Control et SQL*Plus.

18

Gérer l'espace

Laurent PERPETUE (laurent.perpetue@sr.fr) has a non-transferable license to use this Student Guide

Objectifs

A la fin de ce chapitre, vous pourrez :

- décrire comment le serveur de base de données Oracle gère automatiquement l'espace
- économiser l'espace en utilisant la compression
- surveiller et gérer de façon proactive l'utilisation de l'espace dans les tablespaces
- décrire la création de segments dans une base Oracle
- utiliser la création différée de segments
- utiliser la fonction de conseil Segment Advisor
- récupérer l'espace inutilisé des tables et des index à l'aide de la fonctionnalité de récupération d'espace dans les segments
- gérer la reprise après un problème d'allocation d'espace

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Gestion de l'espace : Présentation

Le serveur de base de données Oracle assure automatiquement la gestion de l'espace. Il génère des alertes sur les problèmes potentiels et recommande les solutions possibles. Les fonctionnalités sont les suivantes :

- Oracle Managed Files (OMF)
- Gestion de l'espace libre à l'aide de bitmaps (tablespaces gérés localement) et extension automatique des fichiers de données
- Gestion proactive de l'espace (seuils par défaut et alertes générées par le serveur)
- Récupération d'espace (récupération de l'espace de segments, redéfinition de table en ligne)
- Planification de la gestion de l'espace (états sur la croissance)

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Gestion de l'espace : Présentation

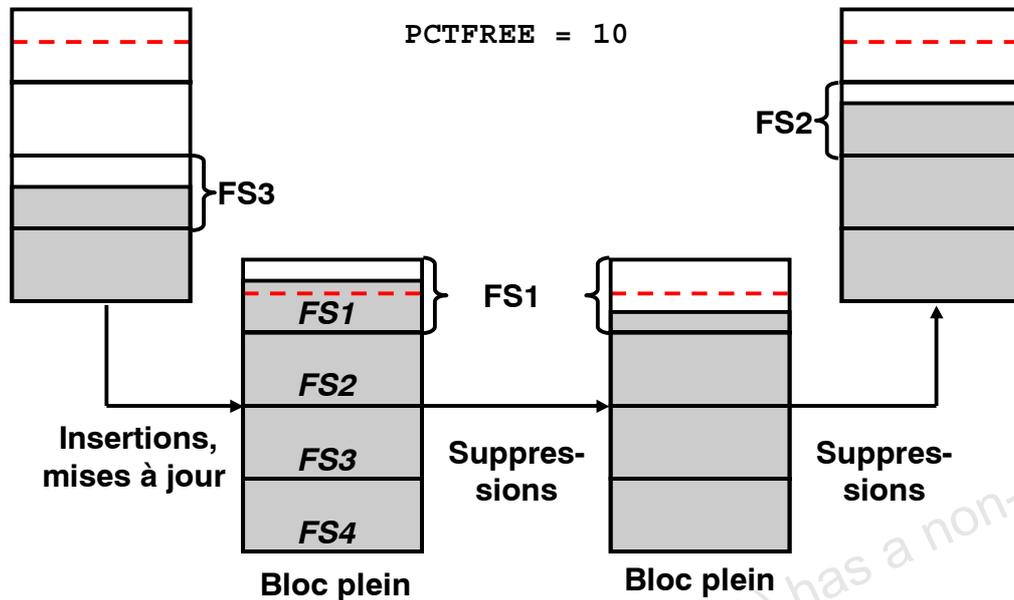
Avec Oracle Managed Files (OMF), vous pouvez indiquer les opérations en termes d'objets de base de données plutôt qu'en termes de fichiers. Le serveur de base de données Oracle peut gérer l'espace libre à l'intérieur d'un tablespace à l'aide de bitmaps. On parle de tablespace "géré localement". En outre, l'espace libre au sein des segments situés dans des tablespaces gérés localement peut être géré à l'aide de bitmaps. On parle alors de gestion automatique de l'espace dans les segments. L'implémentation à l'aide de bitmaps élimine une grande part du réglage des tables lié à l'espace, tout en améliorant les performances lors des pics de charge. En outre, le serveur de base de données Oracle fournit une fonctionnalité d'extension automatique des fichiers de données. Les fichiers peuvent ainsi croître automatiquement en fonction de la quantité de données qu'ils contiennent.

Lorsque vous créez une base, la surveillance proactive de l'espace est activée par défaut. (Ce paramétrage n'affecte pas les performances.) Le serveur de base de données Oracle surveille l'utilisation de l'espace au cours des opérations standard d'allocation et de libération d'espace. Vous recevez un message d'alerte si la quantité d'espace disponible devient inférieure aux seuils prédéfinis (que vous pouvez modifier). Des fonctions de conseil (advisors) et des assistants vous guident dans les opérations de récupération d'espace.

En matière de planification de la gestion de l'espace, le serveur de base de données Oracle fournit des estimations fondées sur la structure des tables et le nombre de lignes, ainsi qu'un état relatif à la croissance des segments fondé sur les données historiques stockées dans le référentiel AWR (Automatic Workload Repository).

(Ces sujets seront traités dans le présent chapitre et dans le suivant.)

Gestion de l'espace de bloc



Copyright © 2009, Oracle. Tous droits réservés.

ORACLE

Gestion de l'espace de bloc

La gestion de l'espace implique la gestion de l'espace libre au niveau des blocs. Avec la gestion automatique de l'espace dans les segments, chaque bloc est divisé en quatre sections, nommées FS1 (entre 0 et 25 % d'espace libre), FS2 (entre 25 % et 50 % d'espace libre), FS3 (entre 50 % et 75 % d'espace libre) et FS4 (entre 75 % et 100 % d'espace libre).

Le statut de chaque bloc est mis à jour automatiquement en fonction du taux d'espace libre. Vous pouvez ainsi déterminer si un bloc peut être utilisé pour l'insertion d'une ligne spécifique, en fonction de la longueur de celle-ci. Lorsqu'un bloc est "plein", il n'est plus disponible pour de nouvelles insertions.

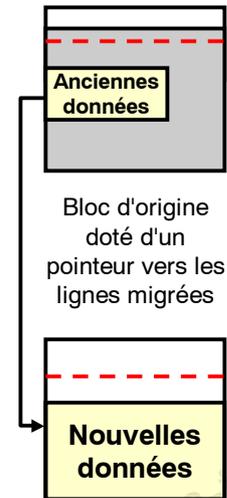
Dans l'exemple de la diapositive ci-dessus, le bloc de gauche est un bloc FS3, car il contient entre 50 % et 75 % d'espace libre. Après l'exécution de plusieurs instructions d'insertion et de mise à jour, la limite $PCTFREE$ est atteinte (ligne en pointillé) et il n'est plus possible d'insérer de nouvelles lignes dans le bloc. Le bloc est alors considéré comme "plein" (statut FS1). Dès que le taux d'espace libre du bloc redescend en dessous de la section suivante, le bloc peut à nouveau être utilisé pour une insertion. Dans le cas de la diapositive ci-dessus, le bloc prend le statut FS2 dès que le taux d'espace libre est supérieur à 25 %.

Remarque : Les objets LOB (BLOB, CLOB, NCLOB et BFILE) n'utilisent pas le paramètre de stockage $PCTFREE$. La valeur $PCTFREE$ par défaut est 10 pour les blocs non compressés et les blocs OLTP compressés, et 0 pour les blocs compressés standard.

Chaînage et migration de lignes

Exemple :

- **Lors d'une mise à jour** : Une ligne augmente en longueur, dépassant l'espace libre disponible dans le bloc.
- Les données doivent être stockées dans un nouveau bloc.
- L'identificateur physique d'origine de la ligne (ROWID) est conservé.
- Le serveur de base de données Oracle doit lire deux blocs pour extraire les données de la ligne.
- Segment Advisor recherche les segments contenant les lignes migrées.
- L'espace libre fragmenté des blocs est fusionné automatiquement.



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Chaînage et migration de lignes

Il existe deux cas de figure dans lesquels les données d'une ligne de table peuvent être trop volumineuses pour tenir dans un bloc unique. Dans le premier cas, la ligne est trop longue pour tenir dans un bloc de données lors de la première insertion. Les données de la ligne sont alors stockées par le serveur de base de données Oracle dans une chaîne de blocs de données (un ou plusieurs) réservée pour ce segment. Le chaînage de lignes se produit la plupart du temps avec les lignes longues, par exemple celles qui contiennent une colonne de type LONG ou LONG RAW. Dans une telle situation, le chaînage de lignes est inévitable.

Dans le second cas, une ligne qui tient initialement dans un bloc de données est mise à jour : la longueur totale de la ligne augmente et l'espace libre du bloc est déjà entièrement rempli. Le serveur migre alors toutes les données de la ligne dans un nouveau bloc, sous réserve que la ligne puisse y tenir intégralement. La base de données conserve le morceau d'origine de chaque ligne migrée pour pointer vers le nouveau bloc contenant la ligne migrée. Le ROWID des lignes migrées est inchangé.

Lorsqu'une ligne fait l'objet d'un chaînage ou d'une migration, les performances d'entrée/de sortie (E/S) associées diminuent car le serveur de base de données Oracle doit balayer plusieurs blocs pour extraire les informations correspondantes.

Segment Advisor recherche les segments qui contiennent des lignes migrées suite à une instruction UPDATE.

Chaînage et migration de lignes (suite)

La base Oracle fusionne automatiquement l'espace libre des blocs de données dans les cas suivants :

- Une instruction INSERT ou UPDATE tente d'utiliser un bloc comprenant suffisamment d'espace libre pour un morceau de ligne.
- L'espace libre est fragmenté de telle manière que le morceau de ligne ne peut pas être inséré dans une section contigüe du bloc.

Après la fusion, l'espace libre a la même taille, mais il est d'un seul tenant.

Quiz

Lorsqu'une ligne fait l'objet d'un chaînage ou d'une migration, les performances d'E/S associées diminuent car la base Oracle doit balayer plusieurs blocs de données pour extraire les informations correspondantes.

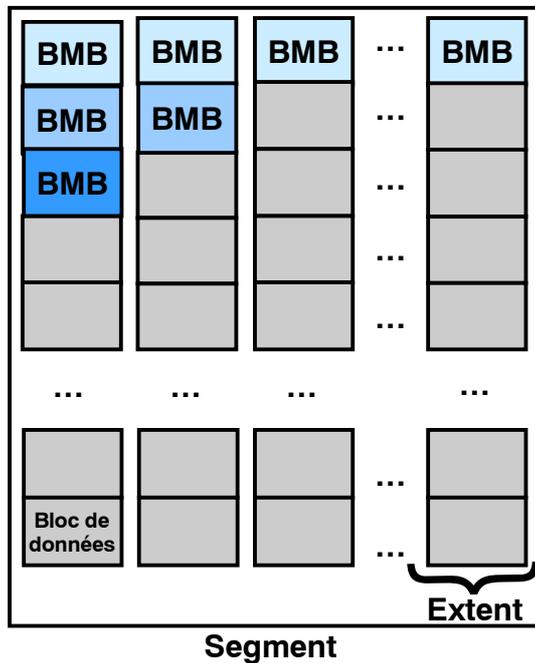
1. Vrai
2. Faux

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Réponse : 1

Gestion de l'espace libre dans des segments



- Suivie par des bitmaps dans les segments

Avantages :

- Utilisation plus souple de l'espace
- Ajustement lors de l'exécution
- Recherche de BMB via plusieurs processus

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Gestion de l'espace libre dans des segments

L'espace libre peut être géré automatiquement à l'intérieur des segments de base de données. Le suivi de l'espace libre ou utilisé dans les segments s'effectue par le biais de bitmaps. Lorsque vous créez un tablespace géré localement, activez la gestion automatique de l'espace dans les segments afin de bénéficier de cette fonctionnalité. Celle-ci s'applique alors à tous les segments créés par la suite dans le tablespace.

Un segment soumis à la gestion automatique de l'espace comporte un ensemble de blocs bitmap (BMB) décrivant l'utilisation de l'espace dans ses blocs de données. Les BMB sont organisés de manière arborescente. Le niveau racine de la hiérarchie, qui contient les références de tous les BMB intermédiaires, est stocké dans l'en-tête du segment.

Les feuilles représentent les informations relatives à l'espace d'un ensemble de blocs de données contigus appartenant au segment. Le nombre maximum de niveaux à l'intérieur de la hiérarchie est de trois.

Les avantages de l'utilisation de la gestion automatique de l'espace sont les suivants :

- Meilleure utilisation de l'espace, en particulier pour les objets qui peuvent comprendre des lignes de tailles très différentes.
- Meilleur ajustement, lors de l'exécution, aux variations dans les accès simultanés.
- Meilleur comportement multi-instance en termes de performances ou d'utilisation de l'espace.

Tous ces avantages réduisent la charge de travail du DBA.

Types de segment

Un segment est un ensemble d'extents alloués pour une certaine structure logique. Les différents types de segment sont les suivants :

- Segments de table et de cluster
- Segment d'index
- Segment d'annulation
- Segment temporaire

Les segments sont alloués de façon dynamique par le serveur de base de données Oracle.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Types de segment

Segments de table et de cluster : Chaque table non clusterisée a un segment de données. Toutes les données de la table sont stockées dans les extents (ensembles de blocs contigus) de ce segment. Dans le cas d'une table partitionnée, chaque partition comporte un segment de données. Chaque cluster comporte un segment de données. Les données de chaque table du cluster sont stockées à l'intérieur du segment de données du cluster.

Segment d'index : Chaque index comporte un segment d'index qui stocke l'ensemble de ses données. Dans le cas d'un index partitionné, chaque partition comporte un segment d'index.

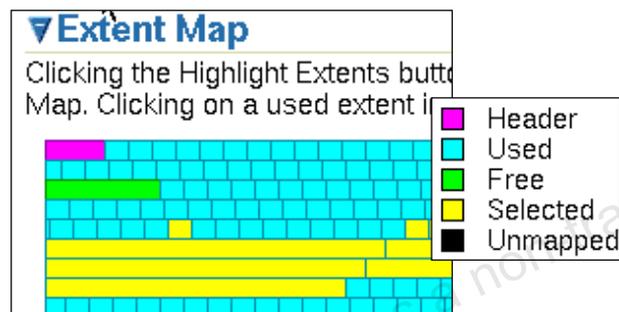
Segment d'annulation : Oracle Database gère des informations permettant d'annuler les modifications apportées à la base de données. Ces informations sont composées d'enregistrements des actions des transactions et sont collectivement appelés informations d'annulation. Les informations d'annulation sont stockées dans des segments d'annulation au sein d'un tablespace d'annulation.

Segment temporaire : Le serveur de base de données Oracle crée un segment temporaire lorsqu'une instruction SQL requiert une zone de travail temporaire pour son exécution. Lorsque l'exécution de l'instruction est terminée, les extents du segment temporaire sont rendus au système en vue d'une utilisation ultérieure.

Le serveur de base de données Oracle alloue dynamiquement de l'espace lorsque les extents existants d'un segment sont pleins. Les extents sont ainsi affectés selon les besoins. Il se peut donc que les extents d'un segment ne soient pas contigus sur le disque.

Allocation d'extents

- Recherche du nombre de blocs libres adjacents requis dans le bitmap du fichier de données
- Dimensionnement des extents à l'aide de clauses de stockage :
 - UNIFORM
 - AUTOALLOCATE
- Affichage de la topographie des ensembles de blocs contigus
- Obtention d'un conseil pour la libération d'espace



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Allocation d'extents

Avec les tablespaces gérés localement, le serveur de base de données Oracle recherche l'espace libre à allouer à un nouvel extent en déterminant d'abord un fichier de données candidat dans le tablespace, puis en recherchant dans le bitmap de ce fichier le nombre de blocs libres adjacents requis. Si le fichier de données n'offre pas suffisamment d'espace libre adjacent, le serveur effectue une recherche dans un autre fichier de données.

Deux clauses affectent le dimensionnement des extents :

- Avec la clause UNIFORM, la base de données crée tous les extents avec la taille que vous indiquez (ou une taille par défaut), pour tous les objets du tablespace.
- Avec la clause AUTOALLOCATE, la base de données détermine la stratégie de dimensionnement des extents pour le tablespace.

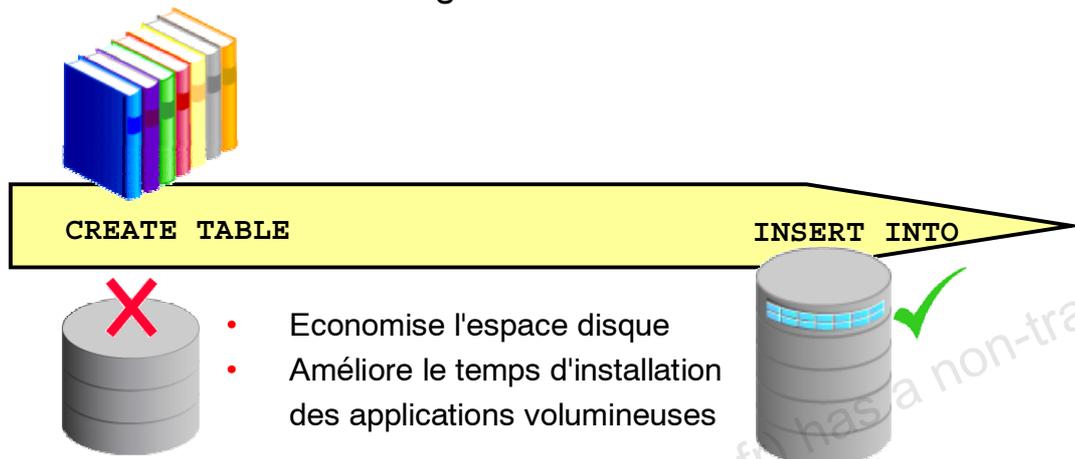
Pour afficher la topographie des ensembles de blocs contigus (extent map) dans Enterprise Manager, sélectionnez Server > Tablespaces > View Tablespace > Show Tablespace Contents.

Le serveur de base de données Oracle fournit la fonction de conseil Segment Advisor, qui vous aide à déterminer si un objet a de l'espace disponible pouvant être récupéré en fonction du niveau de fragmentation.

Allouer de l'espace

Nouvelle méthode d'allocation d'espace :

- DEFERRED_SEGMENT_CREATION = TRUE (par défaut)
1. Création de table > Opération dans le dictionnaire de données
 2. LMD > Création de segment



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Allouer de l'espace

Oracle Database 11gR2 fournit une nouvelle méthode d'allocation d'espace. Lorsque vous créez une table organisée en "heaps" non partitionnée, la création de segments de table est différée jusqu'à l'insertion de la première ligne. Cette fonctionnalité est activée par défaut avec le paramètre d'initialisation DEFERRED_SEGMENT_CREATION défini avec la valeur TRUE.

Avantages de cette nouvelle méthode d'allocation de l'espace :

- Une quantité significative d'espace disque peut être économisée pour les applications qui créent des centaines de milliers de tables dont la plupart ne seront jamais renseignées.
- La durée d'installation de l'application est réduite.

Lorsque vous insérez la première ligne dans la table, les segments sont créés pour la table principale, ses colonnes LOB et ses index. Lors de la création du segment, les curseurs de la table sont invalidés. Ces opérations ont une faible incidence supplémentaire sur les performances.

Remarque : Avec cette nouvelle méthode d'allocation de l'espace, il est essentiel de planifier la capacité appropriée afin que la base de données dispose d'un espace disque suffisant pour assurer la création de segments lorsque les tables sont alimentées. Pour plus d'informations, consultez le manuel *Oracle Database Administrator's Guide*.

Créer des tables sans segments

```
SQL> SHOW PARAMETERS deferred_segment_creation
NAME                                TYPE                                VALUE
-----                                -                                -
deferred_segment_creation            boolean                             TRUE

SQL> CREATE TABLE seg_test(c number, d varchar2(500));
Table created.
SQL> SELECT segment_name FROM user_segments;
no rows selected
```

Insérer des lignes et créer des segments :

```
SQL> INSERT INTO seg_test VALUES(1, 'aaaaaaa');
1 row created.

SQL> SELECT segment_name FROM user_segments;
SEGMENT_NAME
-----
SEG_TEST
```

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Créer des tables sans segments

La diapositive ci-dessus montre comment activer le paramètre DEFERRED_SEGMENT_CREATION. Une table sans segments est alors créée, comme vous pouvez le constater en interrogeant la vue USER_SEGMENTS du dictionnaire de données. Après avoir inséré une ligne, vous interrogez à nouveau cette vue pour vous assurer que le segment existe bien.

Vous pouvez également interroger la colonne SEGMENT_CREATED des vues USER_TABLES, USER_INDEXES ou USER_LOBS. Pour les tables non partitionnées, les index et les objets LOB, cette colonne indique YES, si le segment est créé.

Un autre ajout au dictionnaire de données est la table SYS.SEG\$ que vous avez indiquée lors de la création de la table ou de l'index.

Contrôler la création différée de segments

Avec le paramètre `DEFERRED_SEGMENT_CREATION` :

- Fichier d'initialisation
- Commande `ALTER SESSION`
- Commande `ALTER SYSTEM`

Avec la clause `SEGMENT CREATION` :

- `IMMEDIATE`
- `DEFERRED` (valeur par défaut dans Oracle Database 11gR2)

```
CREATE TABLE SEG_TAB3 (C1 number, C2 number)
  SEGMENT CREATION IMMEDIATE TABLESPACE SEG_TBS;
CREATE TABLE SEG_TAB4 (C1 number, C2 number)
  SEGMENT CREATION DEFERRED;
```

Remarque : Les index héritent des caractéristiques de la table.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Contrôler la création différée de segments

La création de segments peut être contrôlée de deux manières :

- Avec le paramètre d'initialisation `DEFERRED_SEGMENT_CREATION` défini avec la valeur `TRUE` ou `FALSE`. Il peut être configuré dans le fichier d'initialisation. Vous pouvez également la contrôler à l'aide des commandes `ALTER SESSION` ou `ALTER SYSTEM`.

Exemples :

```
ALTER SESSION SET DEFERRED_SEGMENT_CREATION = TRUE;
ALTER SYSTEM SET DEFERRED_SEGMENT_CREATION = FALSE;
```

- Avec la clause `SEGMENT CREATION` de la commande `CREATE TABLE` :
 - `SEGMENT CREATION DEFERRED` : La création de segments est différée jusqu'à ce que la première ligne soit insérée dans la table. Il s'agit du comportement par défaut d'Oracle Database 11gR2.
 - `SEGMENT CREATION IMMEDIATE` : Les segments sont matérialisés lors de la création de la table. Il s'agit du fonctionnement par défaut dans les bases de données Oracle antérieures à Oracle Database 11gR2.

Cette clause prévaut sur le paramètre `DEFERRED_SEGMENT_CREATION`.

Il est possible de forcer la création de segments pour une table déjà créée via la commande `ALTER TABLE ... MOVE`.

Il est cependant impossible de contrôler directement la création différée de segments pour des objets dépendants tels que des index. Ils héritent cette caractéristique de leur objet parent, qui est une table dans le cas présent.

Restrictions et exceptions

Création de segments à la demande :

- Uniquement pour les tables et les index non partitionnés
- Ne s'applique pas aux tables organisées en index, aux tables clusterisées et aux autres tables spéciales
- Ne s'applique pas aux tables figurant dans des tablespaces gérés au moyen d'un dictionnaire

Remarque : Si vous souhaitez migrer une table sans segments d'un tablespace géré localement vers un tablespace géré par un dictionnaire, vous devez la supprimer puis la recréer.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Restrictions et exceptions

- Dans Oracle Database 11gR2, la création différée de segments est restreinte aux tables et aux index non partitionnés. Les tables organisées en index et les autres tables spéciales ne sont pas prises en charge.
- La création de segments à la demande n'est pas possible pour les tables créées dans des tablespaces gérés par un dictionnaire et pour les tables clusterisées. Une telle tentative crée des segments.
- Si vous créez une table via la méthode de création différée de segments dans un tablespace géré localement, elle ne possède aucun segment. Si, par la suite, vous décidez de migrer ce tablespace vers un tablespace géré par un dictionnaire, toute tentative de création de segments génère des erreurs. Dans ce cas, vous devez supprimer la table et la recréer.
- La création de segments à la demande n'est pas prise en charge pour les tables clusterisées, les tables temporaires globales, les tables temporaires spécifiques d'une session, les tables internes, les tables typées, les tables AQ, les tables dont le propriétaire est SYS, les tables externes, les index de jointure bitmap et les index de domaine. Les tables détenues par SYSTEM, PUBLIC, OUTLN et XDB sont également exclues.

Fonctionnalités automatiques supplémentaires

Sans intervention de l'utilisateur :

- Pas de segments pour les index de partitions et les index inutilisables
- Créer un index sans segment :

```
CREATE INDEX test_i1 ON seg_test(c) UNUSABLE;
```

- Supprimer tout espace alloué pour un index :

```
ALTER INDEX test_i UNUSABLE;
```

- Créer le segment pour un index :

```
ALTER INDEX test_i REBUILD;
```

```
SELECT segment_name, partition_name,  
       segment_type  
FROM   user_segments  
WHERE  segment_name like '%DEMO';
```

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Fonctionnalités automatiques supplémentaires

Oracle Database 11g Release 2 introduit d'autres améliorations (non liées à la création différée de segments) pour économiser l'espace. Les index et les partitions d'index UNUSABLE sont créés sans segment. Cette fonctionnalité est entièrement transparente. Elle est activée par défaut avec le paramètre d'initialisation COMPATIBILITY défini avec la valeur 11.2.0.0.

Exemple : Vous possédez une table nommée DEMO comprenant trois partitions et un index local. Vous voyez trois segments de table et trois segments d'index lors de l'exécution de l'interrogation de la diapositive.

Si vous exécutez la même interrogation après avoir déplacé une partition de table vers un nouveau tablespace, vous voyez trois segments de table et seulement deux segments d'index, car l'index inutilisable est automatiquement supprimé.

Quiz

Parmi les affirmations suivantes, lesquelles sont vraies pour Oracle Database 11g Release 2 ?

1. La création différée de segments est toujours activée. Vous ne pouvez pas la contrôler.
2. Vous pouvez contrôler la création différée de segments à l'aide de la clause `SEGMENT CREATION` de la commande `CREATE TABLE`.
3. La création de segments à la demande est disponible pour tous les types de table, y compris pour celles qui sont détenues par l'utilisateur `SYS`.
4. La création de segments à la demande est disponible pour les tables non partitionnées.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Réponses : 2, 4

Compression de table : Présentation

Réduire les coûts de stockage en compressant toutes les données :

- Compression standard pour les insertions par chemin direct : **10x**
- Compression OLTP pour toutes les opérations LMD : **2–4x**

```
{ COMPRESS [ BASIC | FOR { OLTP } ] | NOCOMPRESS }
```

Méthode de compression	Taux de compression	Surcharge de la CPU	Syntaxe CREATE et ALTER TABLE	Applications standard
Compression de table standard	Elevé	Minimale	COMPRESS [BASIC]	DSS
Compression de table OLTP	Elevé	Minimale	COMPRESS FOR OLTP	OLTP, DSS



Copyright © 2009, Oracle. Tous droits réservés.

Compression de table : Présentation

Oracle Database prend en charge trois méthodes de compression de table :

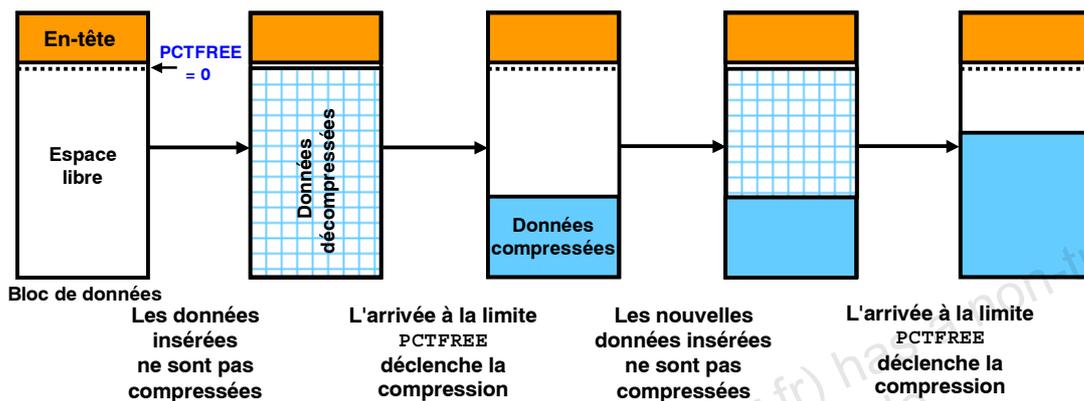
- Compression de table standard
- Compression de table OLTP
- Compression de colonne hybride (avec Exadata)

Oracle Corporation recommande de compresser toutes les données afin de réduire les coûts de stockage. Oracle Database peut utiliser la compression de table pour éliminer les valeurs en double d'un bloc de données. Pour les tables dont les données sont très redondantes, la compression permet d'économiser l'espace disque et de réduire la mémoire utilisée dans le cache de tampons de la base. La compression de table est transparente pour les applications de base de données.

- La clause `TABLE_COMPRESSION` est uniquement valide pour les tables organisées en "heaps". Le mot-clé `COMPRESS` active la compression de table. Le mot-clé `NOCOMPRESS` la désactive. La valeur par défaut est `NOCOMPRESS`.
- Avec la compression standard, Oracle Database compresse les données lors de chargements en masse utilisant des opérations telles que les chargements directs ou `CREATE TABLE AS SELECT`.
- Avec l'option `COMPRESS FOR OLTP`, Oracle Database compresse les données lors de chaque opération LMD effectuée sur la table.

Compression pour les insertions par chemin direct

- Activée par `CREATE TABLE ... COMPRESS BASIC ...;`
- Recommandée pour les data warehouses utilisant le chargement en masse
- Remplace l'option en voie d'abandon `COMPRESS FOR DIRECT_LOAD OPERATIONS`
- Favorise la présence d'espace libre contigu dans les blocs



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Compression pour les insertions par chemin direct

Vous activez la compression standard à l'aide de l'option `COMPRESS` ou `COMPRESS BASIC`.

- Oracle Database tente de compresser les données au cours des opérations suivantes d'insertion par chemin direct, lorsque cela est productif :
 - `SQL*Loader`
 - Instructions `CREATE TABLE AS SELECT`
 - Instructions `INSERT` parallèles
 - Instructions `INSERT` avec le conseil `APPEND`
- L'utilitaire d'import d'origine (`imp`) ne prend pas en charge les opérations `INSERT` par chemin direct. Il ne peut donc pas importer les données sous forme compressée.
- Dans les versions précédentes, ce type de compression était appelé compression de table DSS et était activé via l'option `COMPRESS FOR DIRECT_LOAD OPERATIONS`. Cette syntaxe n'est plus utilisée.
- La compression élimine les trous créés suite à des suppressions et optimise l'espace libre contigu dans les blocs.

La diapositive ci-dessus présente l'évolution d'un bloc de données qui appartient à une table compressée. Consultez-la de gauche à droite. Au début, le bloc est vide et disponible pour les insertions. Lorsque vous commencez à insérer des données dans le bloc, celles-ci sont stockées dans un format non compressé (comme pour les tables non compressées). Cependant, dès que vous atteignez la limite `PCTFREE` du bloc, les données sont automatiquement compressées, ce qui permet de réduire l'espace occupé.

Compression pour les insertions par chemin direct (suite)

Il est ainsi possible d'effectuer de nouvelles insertions de données non compressées dans le bloc, jusqu'à ce que la limite PCTFREE soit à nouveau atteinte. A ce stade, la compression est à nouveau déclenchée pour réduire la quantité d'espace utilisée dans le bloc.

Remarque : Les tables associées à l'option COMPRESS ou COMPRESS BASIC utilisent une valeur PCTFREE de 0 pour optimiser la compression, sauf si vous définissez explicitement une autre valeur.

Les tables associées à l'option COMPRESS FOR OLTP ou NOCOMPRESS utilisent une valeur PCTFREE par défaut égale à 10 pour optimiser la compression tout en autorisant d'autres modifications LMD sur les données, sauf si vous définissez explicitement une autre valeur.

Compression OLTP pour les opérations LMD

- Activée avec
CREATE TABLE ... COMPRESS FOR OLTP ...;
- Recommandée pour les environnements OLTP actifs
- Remplace l'option en voie d'abandon COMPRESS FOR ALL OPERATIONS

	Y		Y		Y
G		Y		G	
	G		Y	Y	G

Bloc non compressé

G	Y				
	Y		Y		Y
G		Y		G	
	G		Y	Y	G

Compression OLTP avec la table de symboles au début du bloc

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Compression OLTP pour les opérations LMD

Vous activez la compression de table OLTP à l'aide de l'option COMPRESS FOR OLTP.

- Oracle Database compresse les données lors de chaque opération LMD effectuée sur la table. Cette forme de compression est recommandée pour les environnements OLTP actifs.
- Dans les version précédentes, la compression de table OLTP était activée avec l'option COMPRESS FOR ALL OPERATIONS. Cette syntaxe n'est plus utilisée.

Avec la compression OLTP, les valeurs en double des lignes et des colonnes d'un bloc de données sont stockées une fois au début du bloc dans une table de symboles. Les valeurs en double sont remplacées par une courte référence à cette table (comme indiqué dans la diapositive). Par conséquent, les informations nécessaires pour recréer les données non comprimées sont stockées dans le bloc.

Pour illustrer le principe de la compression OLTP, le diagramme de la diapositive affiche deux rectangles. Le premier rectangle gris contient quatre petits carrés verts marqués "G" et six carrés jaunes marqués "Y". Ils représentent les blocs non comprimés. Etudions maintenant le deuxième rectangle gris. Au début figurent seulement un carré vert marqué "G" et un carré jaune marqué "Y", qui représentent la table de symboles. Dix carrés blancs sont représentés aux mêmes emplacements que les carrés verts et jaunes. Ils sont seulement indiqués pour référence. Aucun espace n'est consommé par les valeurs en double.

Définir la compression de table

Vous pouvez définir la compression de table pour :

- l'ensemble d'une table organisée en "heaps"
- une table partitionnée (chaque partition peut avoir un type ou un niveau de compression différent)
- le stockage d'une table imbriquée

Vous ne pouvez pas :

- utiliser la compression standard ou OLTP sur les tables comprenant plus de 255 colonnes
- supprimer une colonne si la table est compressée pour les opérations à chargement direct, mais cela est possible avec la compression OLTP

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Définir la compression de table

Vous pouvez définir la compression de table :

- Pour l'ensemble d'une table organisée en "heaps" (dans la clause `PHYSICAL_PROPERTIES<!-- RUN FORMAT="TIMES NEW ROMAN:12:-16777216:B:I:U:S:E:H:T" -->` de `RELATIONAL_TABLE<!-- RUN FORMAT="TIMES NEW ROMAN:12:-16777216:B:I:U:S:E:H:T" -->` ou `OBJECT_TABLE<!-- RUN FORMAT="TIMES NEW ROMAN:12:-16777216:B:I:U:S:E:H:T" -->`).
- Pour les tables partitionnées (chaque partition peut avoir un type ou un niveau de compression différent).
- Pour le stockage d'une table imbriquée (dans la clause `NESTED_TABLE_COL_PROPERTIES`).

La compression de table présente les limites suivantes :

- Les clauses `COMPRESS FOR OLTP` et `COMPRESS BASIC` ne sont pas prises en charge pour les tables contenant plus de 255 colonnes.
- Vous ne pouvez pas supprimer une colonne de table compressée pour des opérations à chargement direct, bien qu'il soit possible de la définir en tant que colonne inutilisée. Toutes les opérations associées à la commande `ALTER TABLE ... DROP_COLUMN_CLAUSE` sont valides pour toutes les tables qui sont compressées pour les opérations OLTP.
- Les bases de secours logiques, Streams et LogMiner ne sont pas pris en charge pour les tables utilisant la compression de colonne hybride.

Utiliser la fonction de conseil sur la compression

La fonction de conseil sur la compression :

- Analyse les objets et évalue les gains d'espace possibles avec différentes méthodes de compression
- Aide à choisir le niveau de compression approprié pour une application
- Recommande différentes stratégies de compression
 - Choisit le bon algorithme pour un jeu de données spécifique
 - Effectue un tri sur une colonne particulière pour augmenter le taux de compression
 - Présente les compromis possibles entre différents algorithmes de compression
- Peut être utilisée pour la compression OLTP (via EM)

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Utiliser la fonction de conseil sur la compression

Cette fonction analyse les objets de la base de données et évalue les taux de compression pouvant être atteints pour chaque niveau de compression. Elle vous aide ainsi à choisir les niveaux de compression appropriés pour votre application. Elle recommande différentes stratégies de compression. Lorsque vous y accédez à partir d'EM, elle détermine la compression OLTP.

Utiliser le package DBMS_COMPRESSION

Pour déterminer les meilleurs taux de compression :

```
BEGIN
DBMS_COMPRESSION.GET_COMPRESSION_RATIO ('USERS','SH','SALES',
NULL,DBMS_COMPRESSION.COMP_FOR_OLTP, blkcnt_cmp, blkcnt_uncmp,
rowcnt_cmp, rowcnt_uncmp, comptype);
DBMS_OUTPUT.PUT_LINE('Blk count compressed = ' || blkcnt_cmp);
DBMS_OUTPUT.PUT_LINE('Blk count uncompressed = ' ||
blkcnt_uncmp);
DBMS_OUTPUT.PUT_LINE('Row count per block compressed = ' ||
rowcnt_cmp);
DBMS_OUTPUT.PUT_LINE('Row count per block uncompressed = ' ||
rowcnt_uncmp);
DBMS_OUTPUT.PUT_LINE('Compression type = ' || comptype);
DBMS_OUTPUT.PUT_LINE('Compression ratio =
' || blkcnt_uncmp/blkcnt_cmp || ' to 1');
```

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Utiliser le package DBMS_COMPRESSION

Une fonction de conseil sur la compression, fournie par le package DBMS_COMPRESSION, vous aide à déterminer le taux de compression pouvant être obtenu pour une table spécifique. Cette fonction analyse les objets de la base de données, détecte les taux de compression pouvant être atteints et recommande les niveaux de compression optimaux. Elle peut aussi être utilisée dans l'environnement de conseil existant (package DBMS_ADVISOR).

Pour déterminer le taux de compression, le package DBMS_COMPRESSION utilise les sous-programmes suivants :

- La procédure GET_COMPRESSION_RATIO donne le taux de compression possible pour une table non compressée.
- La procédure GET_COMPRESSION_TYPE renvoie le type de compression pour une ligne donnée.

Pour plus d'informations, reportez-vous au manuel *Oracle Database PL/SQL Packages and Types Reference*.

Compression des données d'une table

Méthode de compression	Taux de compression	Surcharge de la CPU	Syntaxe CREATE et ALTER TABLE	Applications standard
Compression de table standard	Elevé	Minimale	COMPRESS [BASIC]	DSS
Compression de table OLTP	Elevé	Minimale	COMPRESS FOR OLTP	OLTP, DSS

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Compression des données d'une table

Oracle Database prend en charge trois méthodes de compression de table :

- Compression de table standard
- Compression de table OLTP
- Compression de colonne hybride (avec Exadata)

Surveillance proactive des tablespaces

The screenshot displays the Oracle Enterprise Manager 11g interface for monitoring tablespaces. The main window shows a list of tablespaces with columns for Name, Allocated Size, Space Used, Allocated Space Used, Allocated Free Space, Status, Datafiles, Type, Extent Management, and Segment Management. A table with one row (EXAMPLE) is visible. Below this, the 'Edit Tablespace: EXAMPLE' dialog is open, showing the 'Thresholds' tab. This tab contains two sections: 'Space Used (%)' and 'Free Space (MB)'. Each section has radio buttons for 'Use Database Default Thresholds' and 'Specify Thresholds'. Under 'Specify Thresholds', there are input fields for 'Warning (%)' and 'Critical (%)' for both sections. The 'Space Used (%)' section has values of 85 for Warning and 97 for Critical. The 'Free Space (MB)' section has 'Not Defined' for both Warning and Critical. The 'Options' section has a 'Compress' option set to 'No'.

Copyright © 2009, Oracle. Tous droits réservés.

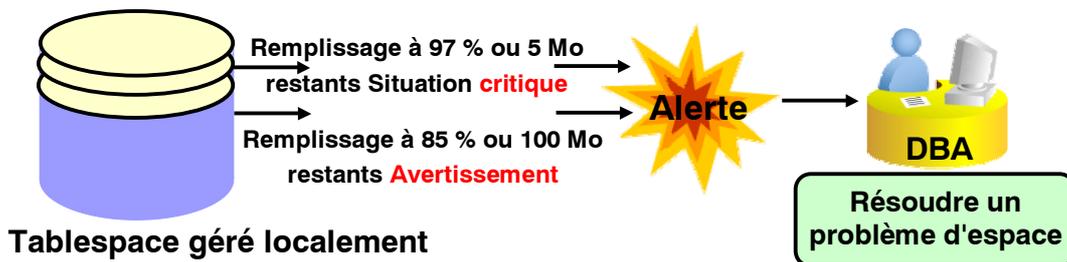
Surveillance proactive des tablespaces

L'utilisation de l'espace disque dans les tablespaces est gérée de façon proactive par la base de données de la manière suivante :

- Via l'utilisation d'alertes de base de données, vous êtes informé lorsqu'un tablespace manque d'espace disque disponible, ainsi que lorsque des segments particuliers manquent d'espace. Vous pouvez alors allouer davantage d'espace disque au tablespace, évitant ainsi les situations de manque d'espace.
- Les informations collectées sont stockées dans le référentiel AWR (Automatic Workload Repository). Elles sont utilisées pour analyser les tendances de croissance et pour planifier la capacité de la base de données.

Pour consulter et modifier les informations sur les tablespaces dans Enterprise Manager, sélectionnez Server dans la page d'accueil de la base de données, puis sélectionnez Tablespaces. Sélectionnez le tablespace souhaité, puis cliquez sur le bouton Edit.

Seuils et résolution de problèmes d'espace



Vous pouvez résoudre un problème d'espace :

- en ajoutant ou redimensionnant un fichier de données
- en activant AUTOEXTEND
- en récupérant l'espace de certains objets
- en réduisant UNDO_RETENTION
- en vérifiant les interrogations longues dans les tablespaces temporaires

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Seuils et résolution de problèmes d'espace

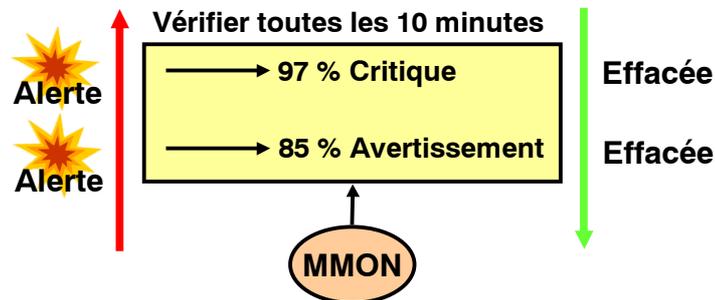
Les seuils associés à un tablespace peuvent être définis sous la forme d'un taux de remplissage ou sous la forme d'une quantité d'espace disponible. Deux types de seuil peuvent être appliqués à un tablespace : le seuil critique et le seuil d'avertissement. Le package DBMS_SERVER_ALERT contient des procédures qui permettent de définir ces seuils et d'extraire les valeurs correspondantes. Lorsque la limite associée à un tablespace est atteinte, l'alerte appropriée est générée. Un seuil peut être exprimé sous la forme d'un pourcentage de la taille du tablespace ou sous la forme d'une valeur correspondant au nombre d'octets disponibles restants. Il est calculé en mémoire. Il est possible de définir à la fois un seuil basé sur un pourcentage et un seuil basé sur un nombre d'octets pour un même tablespace. Dans ce cas, les alertes peuvent être générées par l'un ou l'autre de ces seuils, ou par les deux.

Définissez la valeur de déclenchement du seuil d'avertissement de manière à ce que l'alerte générée soit suffisamment précoce (pour permettre la résolution du problème avant qu'il ne devienne critique), mais pas trop (pour éviter les notifications intempestives).

L'alerte indique que le problème peut être résolu en effectuant une ou plusieurs des opérations suivantes :

- Accroissement de l'espace dans le tablespace en ajoutant un fichier ou en redimensionnant les fichiers existants, ou en activant l'attribut d'auto-extension AUTOEXTEND pour un fichier existant.
- Libération d'espace sur les disques contenant des fichiers en auto-extension.
- Récupération d'espace dans le tablespace par regroupement des objets dispersés (shrink).

Surveillance de l'utilisation de l'espace dans les tablespaces



- Tablespaces en lecture seule et tablespaces hors ligne : Ne configurez pas d'alertes.
- Tablespace temporaire : Le seuil correspond à l'espace actuellement utilisé par les sessions.
- Tablespace d'annulation : Le seuil correspond à l'espace utilisé par les extents actifs et n'ayant pas expiré.
- Fichiers en auto-extension : Le seuil est basé sur la taille maximale des fichiers.

ORACLE

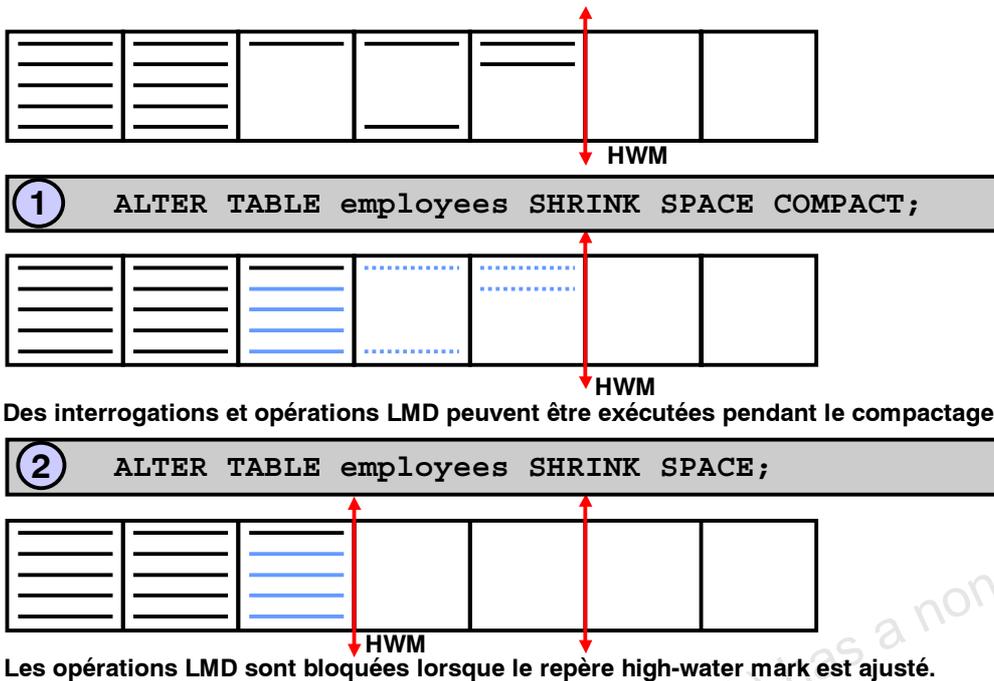
Copyright © 2009, Oracle. Tous droits réservés.

Surveillance de l'utilisation de l'espace dans les tablespaces

La base de données effectue le suivi de l'utilisation de l'espace en procédant à des activités régulières de gestion de l'espace. Ces informations sont agrégées toutes les 10 minutes par le processus MMON. Une alerte est déclenchée lorsque la valeur dépasse le seuil d'un tablespace ou revient à un niveau inférieur.

- Les alertes ne doivent pas être déclenchées pour les tablespaces en lecture seule ou pour les tablespaces qui ont été mis hors ligne (offline), car les possibilités d'action sont limitées pour ces tablespaces.
- Dans les tablespaces temporaires, la valeur de seuil doit être définie comme une limite concernant l'espace utilisé dans le tablespace.
- S'agissant des tablespaces d'annulation, un extent (ensemble de blocs contigus) peut être réutilisé s'il ne contient pas d'informations d'annulation actives ou n'ayant pas expiré. Pour le calcul de la violation des seuils, la somme de l'espace occupé par les extents actifs et n'ayant pas expiré est considérée comme l'espace utilisé.
- Pour les tablespaces avec des fichiers en auto-extension, les seuils sont calculés en fonction de la taille maximale que vous avez définie pour les fichiers, ou de la taille maximale des fichiers du système d'exploitation.

Récupération d'espace dans les segments



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

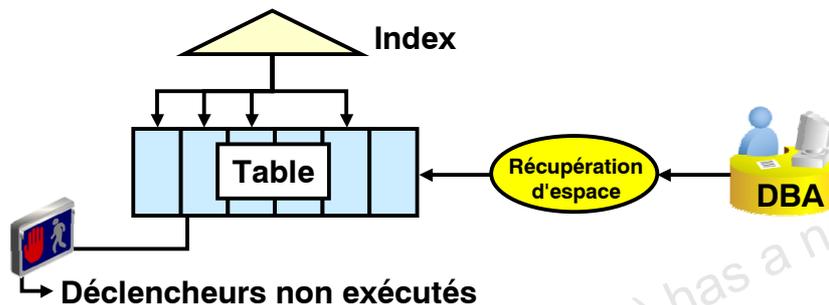
Récupération d'espace dans les segments

Le diagramme de la diapositive ci-dessus décrit les deux phases d'une opération de récupération d'espace (shrink) dans une table. La première phase correspond au compactage. Au cours de cette phase, les lignes sont déplacées vers la partie gauche du segment dans la mesure du possible. En interne, les lignes sont déplacées par paquets afin d'éviter les problèmes de verrouillage. Une fois les lignes déplacées, la deuxième phase de l'opération de récupération d'espace est démarrée. Au cours de cette phase, le repère high-water mark (HWM) est ajusté et l'espace inutilisé est libéré.

La clause COMPACT est utile si des interrogations longues sont concernées par l'opération de récupération d'espace et que vous tentez de lire des blocs dont l'espace a été récupéré. Lorsque vous indiquez la clause SHRINK SPACE COMPACT, la progression de l'opération de récupération d'espace est enregistrée dans les blocs bitmap du segment correspondant. Cela signifie que lors de la prochaine exécution d'une telle opération sur le même segment, le serveur de base de données Oracle mémorise ce qui a déjà été fait. Vous pouvez alors réexécuter la clause SHRINK SPACE sans la clause COMPACT pendant les heures de faible activité, afin de terminer la deuxième phase.

Résultats d'une opération de récupération d'espace

- Les performances et l'utilisation de l'espace sont améliorées.
- Les index sont préservés.
- Déclencheurs non exécutés
- Le nombre de lignes migrées peut être réduit.
- Il est recommandé de reconstruire les index secondaires sur les tables organisées en index.



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Résultats d'une opération de récupération d'espace

Lorsque les segments sont remplis de manière dispersée, la récupération d'espace permet d'améliorer les performances des balayages et des opérations LMD sur ces segments. En effet, après cette opération, le nombre de blocs à examiner est plus limité. Cela est particulièrement vrai pour :

- les balayages complets de table (full table scans), car les blocs sont moins nombreux et plus denses,
- l'accès aux index, car le nombre d'E/S liées aux balayages de ROWID par intervalle sont moins nombreux, grâce à une arborescence plus compacte.

En outre, en procédant à une récupération d'espace dans les segments remplis de manière dispersée, vous renforcez l'efficacité de l'utilisation de l'espace dans la base de données car davantage d'espace libre est mis à la disposition des objets qui en ont besoin.

La dépendance des index est assurée pendant l'opération de récupération d'espace dans les segments. Les index sont dans un état utilisable après la récupération d'espace dans la table correspondante. Par conséquent, aucune maintenance supplémentaire n'est requise.

L'opération proprement dite de récupération d'espace est gérée en interne sous la forme d'une opération INSERT/DELETE. Cependant, les déclencheurs (triggers) LMD ne sont pas exécutés, car les données proprement dites ne sont pas modifiées.

Suite à l'opération de récupération d'espace, il se peut que le nombre de lignes migrées soit réduit. Mais ce n'est pas toujours le cas. En effet, il se peut que l'opération de récupération d'espace ne concerne pas tous les blocs du segment. Par conséquent, il n'est pas possible de garantir que toutes les lignes migrées seront traitées.

Remarque : Il est recommandé de reconstruire les index secondaires sur une table organisée en index après une opération de récupération d'espace.

Récupérer de l'espace dans des segments ASSM

- Opérations en ligne et sur place.
- Applicable uniquement aux segments qui résident dans des tablespaces ASSM.
- Types de segment candidats :
 - Tables organisées en "heaps" et tables organisées en index
 - Index
 - Partitions et sous-partitions
 - Vues matérialisées et journaux des vues matérialisées

ORACLE

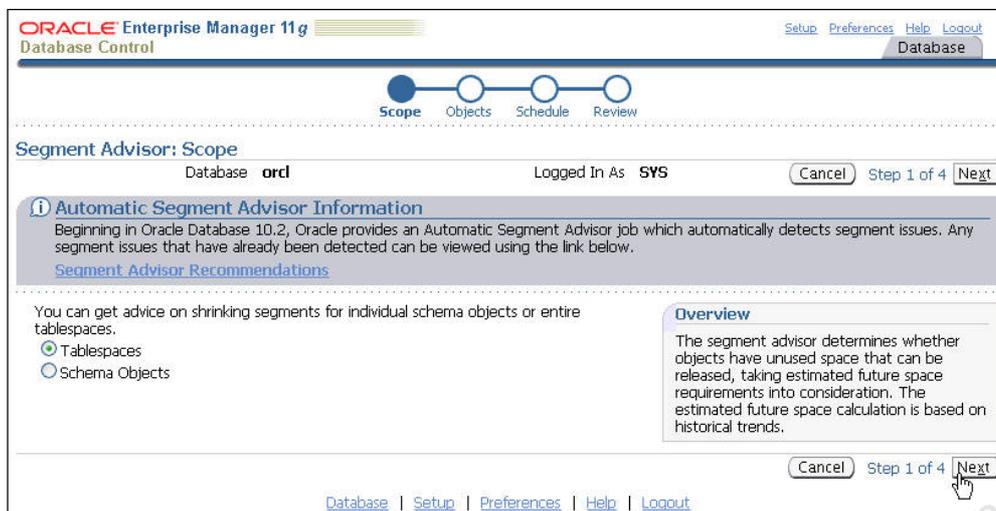
Copyright © 2009, Oracle. Tous droits réservés.

Récupérer de l'espace dans des segments ASSM

Une opération de récupération d'espace est une opération en ligne et sur place, car son exécution ne nécessite pas d'espace de base de données supplémentaire.

- Vous ne pouvez pas exécuter une opération de récupération d'espace sur des segments gérés par des listes de blocs libres (free lists). Les segments qui figurent dans des tablespaces utilisant la gestion automatique de l'espace dans les segments peuvent faire l'objet d'une récupération d'espace. En revanche, les objets suivants stockés dans des tablespaces ASSM ne peuvent pas faire l'objet d'une récupération d'espace :
 - Tables en clusters
 - Tables avec des colonnes LONG
 - Tables avec des vues matérialisées on-commit
 - Tables avec des vues matérialisées basées sur le ROWID
 - Tables de mise en correspondance (mapping tables) IOT
 - Tables avec index basés sur une fonction
- Le déplacement de lignes (`row movement`) doit être activé pour les segments organisés en "heaps".

Segment Advisor : Présentation



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Segment Advisor : Présentation

La fonction de conseil Segment Advisor identifie les segments disposant d'espace libre pouvant être récupéré. Elle effectue son analyse en examinant les statistiques d'utilisation et de croissance dans le référentiel AWR (Automatic Workload Repository), et en échantillonnant les données du segment. Elle est configurée pour s'exécuter automatiquement à intervalles réguliers, mais vous pouvez aussi l'exécuter à la demande (manuellement). L'exécution régulière planifiée de la fonction de conseil Segment Advisor est appelée Automatic Segment Advisor.

Une fois les recommandations effectuées, vous pouvez choisir de les implémenter. La fonction de conseil Segment Advisor peut être appelée au niveau segment ou tablespace.

La console EM Database Control est l'interface de Segment Advisor. Vous pouvez accéder à la fonction de conseil Segment Advisor à partir de différents emplacements dans EM :

- Page Advisor Central
- Page Tablespaces
- Pages des objets de schéma

La console Database Control vous permet de sélectionner différentes entrées et de planifier un travail qui appelle Segment Advisor afin d'obtenir un conseil concernant la récupération d'espace. L'assistant Segment Advisor Wizard peut être appelé sans contexte, dans le contexte d'un tablespace ou dans le contexte d'un objet de schéma.

Les recommandations émises par Segment Advisor sont fondées sur l'analyse par échantillonnage, sur les informations historiques et sur les tendances de croissance futures.

Segment Advisor

The screenshot displays two overlapping windows from Oracle Enterprise Manager 11g Database Control. The top window is titled 'Tablespaces: Add' and shows a list of available tablespaces. The bottom window is titled 'Segment Advisor: Tablespaces' and shows a table of tablespace details. A red arrow points from the 'OK' button in the top window to the 'Submit' button in the bottom window.

Select	Name	Type	Extent Management	Segment Space Management	Size (MB)	Used (MB)	Used (%)
<input type="checkbox"/>	SYSTEM	PERMANENT	LOCAL	MANUAL	710.00	701.56	98.81
<input type="checkbox"/>	TEMP	TEMPORARY	LOCAL	MANUAL	54.00	53.00	98.15
<input type="checkbox"/>	SYSAUX	PERMANENT	LOCAL	AUTO	796.69	757.88	95.13
<input checked="" type="checkbox"/>	EXAMPLE	PERMANENT	LOCAL	AUT			
<input type="checkbox"/>	UNDOTBS1	UNDO	LOCAL	MAN			
<input type="checkbox"/>	USERS	PERMANENT	LOCAL	AUT			

Name	Type	Extent Management	Segment Space Management	Size (MB)	Used (MB)	Used (%)	Remove
EXAMPLE	PERMANENT	LOCAL	AUTO	100.00	77.38	77.38	

Copyright © 2009, Oracle. Tous droits réservés.

Segment Advisor

Dans la page Server, sélectionnez Tablespaces dans la région Storage. Dans la page Tablespaces, sélectionnez le tablespace sur lequel vous souhaitez exécuter l'analyse de récupération d'espace, puis sélectionnez Run Segment Advisor dans la liste déroulante Actions. Cliquez sur Go afin d'ouvrir la page initiale de Segment Advisor. Vous devez choisir le mode d'analyse Comprehensive ou Limited. En mode "Comprehensive", l'analyse est plus longue, car la fonction de conseil échantillonne les segments afin d'identifier les cibles appropriées.

Cliquez sur Continue afin de répondre aux différentes questions de la fonction de conseil. Au final, vous accédez à la page Segment Advisor: Review, dans laquelle vous pouvez revoir les détails de l'analyse. L'analyse Segment Advisor est exécutée en tant que travail planifié, de sorte que vous pouvez revoir la tâche planifiée à partir de la page Advisor Central. Une fois l'exécution terminée, vous pouvez examiner les recommandations de la fonction de conseil.

Remarque : La fonction de conseil Segment Advisor vous permet de définir la durée de l'analyse, c'est-à-dire de limiter le temps nécessaire à la génération des recommandations. De manière générale, une analyse plus longue donne des résultats plus complets. Les résultats sont stockés dans le référentiel AWR et peuvent être examinés ultérieurement. L'option "Number of days to retain" vous permet d'indiquer au serveur de base de données Oracle la durée pendant laquelle ces résultats doivent être conservés avant leur purge du référentiel AWR.

Implémenter les recommandations

Advisor Central

Advisors | Checkers

Page Refreshed Jul 12, 2007 7:18:54 AM EDT (Refresh)

Advisors

[ADDM](#) [Automatic Undo Management](#) [Data Recovery Advisor](#)
[Memory Advisors](#) [MTTR Advisor](#) [Segment Advisor](#)
[SQL Advisors](#) [SQL Performance Analyzer](#)

Advisor Tasks Change Default Parameters

Search

Select an advisory type and optionally enter a task name to filter the data that is displayed in your results set.

Advisory Type: Segment Advisor Task Name: Advisor Runs: All Status: All Go

By default, the search returns all uppercase matches beginning with the string you entered. To run an exact or case-sensitive match, double quote the search string. You can use the wildcard symbol (%) in a double quoted string.

Results

View Result Delete Actions Re-schedule Go Previous 1-25 of 45 Next 20

Select	Advisory Type	Name	Description	User	Status	Start Time	Duration (seconds)	Expires In (days)
<input checked="" type="checkbox"/>	Segment Advisor	SEGMENT.ADV_5412893	Get shrink advice based on object growth trend	SYS	COMPLETED	Jul 12, 2007 7:13:44 AM	134	30

Select	Schema	Segment	Recommendation	Reclaimable Space (MB)	(MB)	(MB) Type
<input checked="" type="checkbox"/>	SYS	EMPLOYEES1	<input type="button" value="Shrink"/>	alter table "SYS"."EMPLOYEES1"	shrink space	
<input checked="" type="checkbox"/>	SYS	EMPLOYEES2	<input type="button" value="Shrink"/>	alter table "SYS"."EMPLOYEES2"	shrink space	
<input checked="" type="checkbox"/>	SYS	EMPLOYEES3	<input type="button" value="Shrink"/>	alter table "SYS"."EMPLOYEES3"	shrink space	

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Implémenter les recommandations

Une fois que la fonction de conseil Segment Advisor a terminé son travail, vous pouvez afficher les recommandations et les implémenter directement.

Remarque : Avant de récupérer l'espace d'une table organisée en "heaps", vous devez activer le déplacement de lignes (row movement) pour cette table. Vous pouvez pour cela utiliser l'onglet Options de la page Edit Table dans Database Control.

Automatic Segment Advisor

La fonction de conseil Automatic Segment Advisor :

- Est démarrée par un travail du planificateur défini pour s'exécuter dans la fenêtre de maintenance par défaut :
 - Nuits de la semaine, du lundi au vendredi, de 22h00 à 2h00
 - Le samedi et le dimanche, les deux fenêtres commencent à 6h00 et durent 20 heures
- Examine les statistiques de base de données, échantillonne les données des segments, puis sélectionne les objets suivants pour analyse :
 - Tablespaces ayant dépassé un seuil critique ou d'avertissement
 - Segments présentant l'activité la plus importante
 - Segments présentant le taux de croissance le plus élevé

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Automatic Segment Advisor

La fonction Automatic Segment Advisor est démarrée par un travail du planificateur. Ce travail est configuré pour s'exécuter dans la fenêtre de maintenance par défaut. Celle-ci est indiquée dans le planificateur et elle est initialement définie comme suit :

- Nuits de la semaine, du lundi au vendredi, de 22h00 à 2h00 (4 heures par nuit)
- Week-ends, samedi et dimanche à 6h00 du matin et pendant 20 heures par jour

La fonction Automatic Segment Advisor n'analyse pas chaque objet de base de données. Elle examine plutôt les statistiques de base de données, échantillonne les données des segments, puis sélectionne les objets suivants pour analyse :

- Tablespaces ayant dépassé un seuil d'espace critique ou d'avertissement
- Segments présentant l'activité la plus importante
- Segments présentant le taux de croissance le plus élevé

Si un objet est sélectionné pour analyse mais que la fenêtre de maintenance expire avant que la fonction Segment Advisor ne puisse le traiter, il est inclus dans l'exécution suivante de la fonction Automatic Segment Advisor. Vous ne pouvez pas modifier le jeu de tablespaces et de segments que la fonction Automatic Segment Advisor sélectionne pour analyse. Toutefois, vous pouvez activer ou désactiver le travail Automatic Segment Advisor, modifier les horaires de son exécution planifiée ou ajuster son utilisation des ressources système.

Récupération manuelle d'espace dans les segments à l'aide d'EM

The screenshot displays two windows from Oracle Enterprise Manager 11g Database Control. The left window shows the 'Tables' page for the 'orcl' database instance. A search for 'HR' is performed, and the 'EMPLOYEES' table is selected. The 'Actions' menu is open, and 'Shrink Segment' is highlighted. The right window shows the 'Shrink Segment' dialog for 'HR.EMPLOYEES'. The 'Shrink Options' section has 'Compact Segments and Release Space' selected. The 'Segment Selection' section has 'Shrink HR.EMPLOYEES and All Dependent Segments' selected. A table of dependent segments is shown below.

Schema	Segment Name	Type	Tablespace
HR	EMPLOYEES	TABLE	EXAMPLE
HR	EMP_EMAIL_UK	INDEX	EXAMPLE
HR	EMP_EMP_ID_PK	INDEX	EXAMPLE
HR	EMP_DEPARTMENT_IX	INDEX	EXAMPLE
HR	EMP_JOB_IX	INDEX	EXAMPLE
HR	EMP_MANAGER_IX	INDEX	EXAMPLE
HR	EMP_NAME_IX	INDEX	EXAMPLE

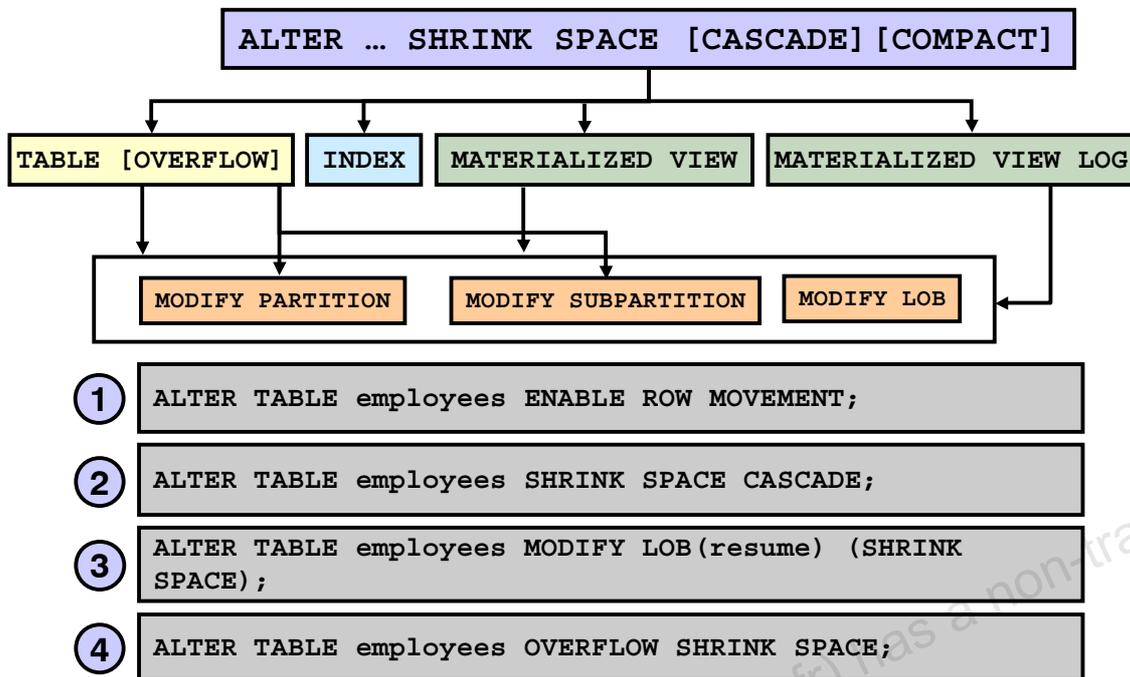
Copyright © 2009, Oracle. Tous droits réservés.

Récupération manuelle d'espace dans les segments à l'aide d'EM

Au lieu d'implémenter les recommandations de la fonction de conseil Segment Advisor, vous pouvez effectuer une récupération d'espace dans différents segments associés à des objets de base de données spécifiques. Par exemple, dans la page d'accueil de la base de données, sélectionnez l'onglet Schema, puis cliquez sur le lien Tables dans la section Database Objects. Dans la page Tables, sélectionnez la table souhaitée, puis sélectionnez Shrink Segment dans la liste déroulante Actions. Cliquez ensuite sur le bouton Go. Vous accédez à la page Shrink Segment, dans laquelle vous pouvez choisir les segments dépendants appropriés. Vous pouvez choisir entre le compactage uniquement ou le compactage et la récupération d'espace. Vous pouvez également sélectionner l'option CASCADE.

Lorsque vous avez terminé, cliquez sur le lien Continue. Les instructions de récupération d'espace sont ainsi soumises en tant que travail planifié.

Récupération d'espace dans les segments à l'aide d'instructions SQL



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Récupération d'espace dans les segments à l'aide d'instructions SQL

Etant donné qu'une opération de récupération d'espace peut entraîner la modification des ROWID dans les segments organisés en "heaps", vous devez activer le déplacement de lignes (row movement) pour le segment correspondant avant d'exécuter une opération de récupération d'espace sur ce segment. Par défaut, le déplacement de lignes est désactivé au niveau segment. Pour l'activer, utilisez la clause `ENABLE ROW MOVEMENT` de la commande `CREATE TABLE` ou `ALTER TABLE`. Cette opération est illustrée dans le premier exemple de la diapositive ci-dessus.

Utilisez la commande `ALTER` sur un objet pour demander une opération de récupération d'espace dans les segments. Le type d'objet peut être l'un des suivants : table (organisée en "heaps" ou en index), partition, sous-partition, index, vue matérialisée ou journal des vues matérialisées.

Utilisez la clause `SHRINK SPACE` pour récupérer l'espace dans un segment. Si l'option `CASCADE` est utilisée, le comportement de l'opération de récupération d'espace est propagé vers tous les segments dépendants prenant en charge une telle opération, à l'exception des vues matérialisées, des index LOB (Large Object) et des tables de mise en correspondance (mapping tables) IOT. La clause `SHRINK SPACE` est illustrée dans le deuxième exemple de la diapositive.

Récupération d'espace dans les segments à l'aide d'instructions SQL (suite)

Dans un segment d'index, l'opération de récupération d'espace fusionne l'index avant de compacter les données.

L'exemple 3 montre une commande qui récupère l'espace dans un segment LOB, à condition que la colonne RESUME soit de type CLOB (Character Large Object).

L'exemple 4 montre une commande qui récupère l'espace dans un segment de débordement (overflow segment) IOT appartenant à la table EMPLOYEES.

Remarque : Pour plus d'informations, reportez-vous au guide *Oracle Database SQL Reference Guide*.

Gérer la reprise après un problème d'allocation d'espace

Une instruction liée au mode de reprise après un problème d'allocation d'espace :

- vous permet de suspendre les opérations volumineuses plutôt que de recevoir une erreur
- vous permet de résoudre le problème pendant que l'opération est suspendue, plutôt que de recommencer
- est suspendue dans les cas suivants :
 - manque d'espace
 - nombre maximal d'extents atteint
 - quota d'espace dépassé
- Une instruction peut faire l'objet d'une mise en suspens et d'une reprise plusieurs fois.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Gérer la reprise après un problème d'allocation d'espace

Le serveur de base de données Oracle offre un moyen de suspendre, puis de reprendre ultérieurement l'exécution d'opérations volumineuses sur la base de données en cas de problème d'allocation d'espace. Cela vous permet de prendre les mesures appropriées, ce qui évite au serveur de base de données Oracle de renvoyer une erreur à l'utilisateur. Une fois la condition d'erreur corrigée, l'exécution de l'instruction mise en suspens reprend automatiquement. Cette fonctionnalité est appelée "reprise après un problème d'allocation d'espace". Les instructions concernées sont appelées "instructions liées au mode de reprise après un problème d'allocation d'espace". Une instruction s'exécute en mode de reprise uniquement lorsque la fonctionnalité de reprise après un problème d'allocation d'espace a été activée pour le système ou la session.

La mise en suspens d'une instruction entraîne automatiquement la mise en suspens de la transaction. Par conséquent, toutes les ressources transactionnelles sont préservées suite à la mise en suspens et à la reprise d'une instruction SQL. Lorsque la situation ayant entraîné l'erreur disparaît (par exemple, suite à l'intervention de l'utilisateur ou une fois que l'espace de tri a été libéré par d'autres interrogations), l'exécution de l'instruction mise en suspens reprend automatiquement. Une instruction liée au mode de reprise est suspendue lorsque l'une des conditions suivantes se produit :

- Manque d'espace
- Nombre maximal d'extents atteint
- Quota d'espace dépassé

Gérer la reprise après un problème d'allocation d'espace (suite)

Un délai d'expiration de mise en suspens est associé aux instructions liées au mode de reprise après un problème d'allocation d'espace. Lorsque la durée de suspension excède ce délai (la valeur par défaut est 2 heures), l'instruction est réactivée et renvoie une exception à l'utilisateur. Une instruction peut faire l'objet d'une mise en suspens et d'une reprise plusieurs fois.

Remarque : L'erreur correspondant au dépassement du nombre maximal d'extents ne peut se produire qu'avec des tablespaces gérés au moyen du dictionnaire.

Utiliser le mode de reprise après un problème d'allocation d'espace

- Les interrogations, les opérations LMD et certaines opérations LDD peuvent faire l'objet d'une reprise si elles rencontrent une erreur de manque d'espace.
- Une instruction liée au mode de reprise après un problème d'allocation d'espace peut être exécutée via SQL, PL/SQL, SQL*Loader, les utilitaires Data Pump ou Oracle Call Interface (OCI).
- Une instruction s'exécute en mode de reprise uniquement si sa session a été activée par l'une des actions suivantes :
 - Le paramètre d'initialisation `RESUMABLE_TIMEOUT` se voit attribuer une valeur différente de zéro.
 - Une instruction `ALTER SESSION ENABLE RESUMABLE` est exécutée :

```
ALTER SESSION ENABLE RESUMABLE;
INSERT INTO sales_new SELECT * FROM sh.sales;
ALTER SESSION DISABLE RESUMABLE;
```

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Utiliser le mode de reprise après un problème d'allocation d'espace

La reprise après un problème d'allocation d'espace n'est possible que lorsque les instructions sont exécutées dans une session pour laquelle le mode de reprise est activé. Il existe deux moyens d'activer et de désactiver le mode de reprise après un problème d'allocation d'espace :

- Exécutez la commande `ALTER SESSION ENABLE RESUMABLE`.
- Affectez au paramètre d'initialisation `RESUMABLE_TIMEOUT` une valeur différente de zéro avec une instruction `ALTER SESSION` ou `ALTER SYSTEM`.

Lors de l'activation du mode de reprise pour une session ou pour la base de données, vous pouvez indiquer une période d'expiration, après laquelle une instruction mise en suspens génère une erreur si aucune intervention n'a eu lieu. Le paramètre d'initialisation `RESUMABLE_TIMEOUT` indique le nombre de secondes avant l'expiration. Vous pouvez également définir cette période à l'aide de la commande suivante :

```
ALTER SESSION ENABLE RESUMABLE TIMEOUT 3600;
```

La valeur du paramètre `TIMEOUT` reste en vigueur tant qu'elle n'est pas modifiée par une autre instruction `ALTER SESSION ENABLE RESUMABLE`, tant qu'elle n'est pas modifiée par un autre moyen ou tant que la session n'est pas terminée. L'intervalle d'expiration par défaut lors de l'utilisation de la clause `ENABLE RESUMABLE TIMEOUT` pour l'activation du mode de reprise est de 7200 secondes, soit 2 heures.

Utiliser le mode de reprise après un problème d'allocation d'espace (suite)

Vous pouvez également nommer les instructions liées au mode de reprise après un problème d'allocation d'espace. Par exemple :

```
ALTER SESSION ENABLE RESUMABLE TIMEOUT 3600
NAME 'multitab insert';
```

Vous pouvez ainsi identifier les instructions dans les vues DBA_RESUMABLE et USER_RESUMABLE.

Par exemple :

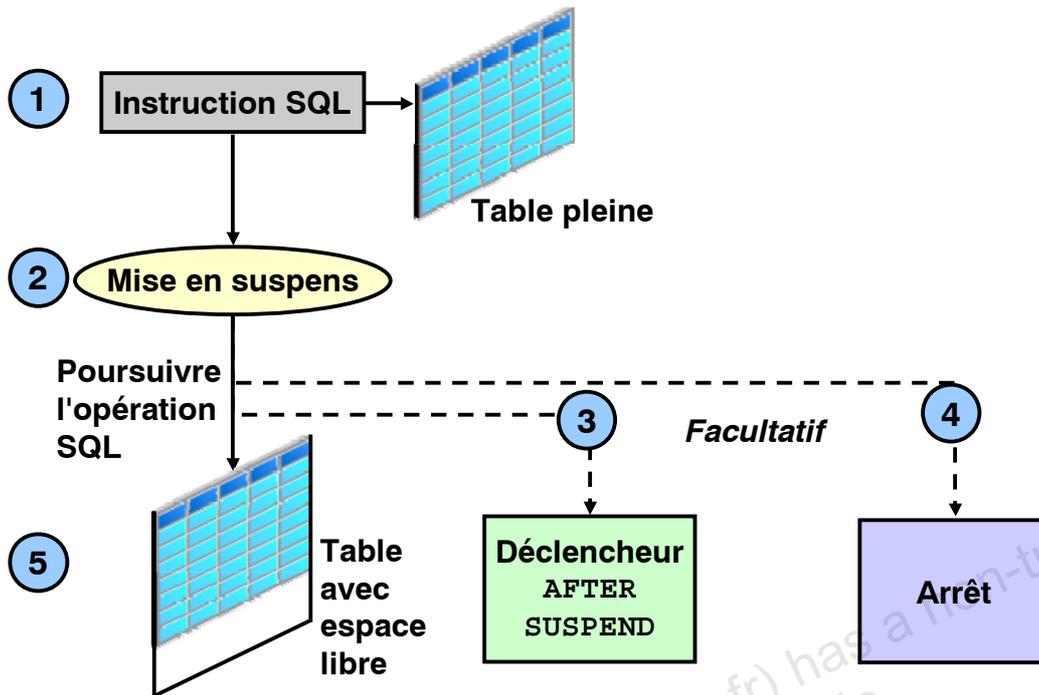
```
SELECT name, sql_text FROM user_resumable;
```

```
NAME          SQL_TEXT
-----
multitab insert INSERT INTO oldsales SELECT * FROM sh.sales;
```

Pour configurer automatiquement les paramètres de reprise après un problème d'allocation d'espace pour des sessions individuelles, vous pouvez créer et enregistrer un déclencheur (trigger) LOGON au niveau de la base de données, qui modifie la session d'un utilisateur. Le déclencheur exécute des commandes afin d'activer les instructions liées au mode de reprise après un problème d'allocation d'espace pour la session, définit une période d'expiration et associe un nom aux instructions exécutées par la session.

Etant donné que les instructions mises en suspens peuvent détenir des ressources système, les utilisateurs doivent disposer du privilège système RESUMABLE pour être autorisés à activer le mode de reprise après un problème d'allocation d'espace et à exécuter les instructions correspondantes.

Reprise d'instructions mises en suspens



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Reprise d'instructions mises en suspens

Exemple

1. Une instruction INSERT donne lieu à une erreur indiquant que la table est pleine.
2. L'instruction INSERT est suspendue et l'erreur n'est pas transmise au client.
3. De manière facultative, un déclencheur (trigger) AFTER SUSPEND est exécuté.
4. Ou bien, l'exception SQLERROR est activée pour arrêter l'instruction.
5. Si l'instruction n'est pas arrêtée et que de l'espace libre est ajouté avec succès à la table, l'exécution de l'instruction INSERT reprend.

Détecter une instruction mise en suspens

Lorsqu'une instruction est mise en suspens, l'erreur n'est pas signalée au client. Pour que l'action corrective soit effectuée, le serveur de base de données Oracle offre des méthodes alternatives pour informer les utilisateurs de l'erreur et pour fournir des informations sur les circonstances.

Reprise d'instructions mises en suspens (suite)

Actions possibles durant la mise en suspens

Lorsqu'une instruction liée au mode de reprise après un problème d'allocation d'espace rencontre une erreur pouvant être corrigée, le système génère en interne l'événement système AFTER SUSPEND. Les utilisateurs peuvent enregistrer des déclencheurs (triggers) pour cet événement au niveau base de données et au niveau schéma. Si un utilisateur enregistre un déclencheur pour gérer cet événement système, le déclencheur est exécuté après la mise en suspens d'une instruction SQL. Les instructions SQL exécutées dans un déclencheur AFTER SUSPEND ne peuvent jamais faire l'objet d'une reprise et sont toujours autonomes. Les transactions démarrées dans le déclencheur utilisent le segment d'annulation (rollback segment) SYSTEM. Ces conditions sont imposées pour faire face aux "verrous mortels" (deadlocks) et pour limiter le risque que le déclencheur soit confronté à la même condition d'erreur que l'instruction.

Dans le code du déclencheur, vous pouvez utiliser la vue USER_RESUMABLE ou DBA_RESUMABLE, ou la fonction DBMS_RESUMABLE.SPACE_ERROR_INFO, pour obtenir des informations sur les instructions liées au mode de reprise après un problème d'allocation d'espace.

Lorsqu'une instruction liée au mode de reprise après un problème d'allocation d'espace est mise en suspens :

- La session qui appelle l'instruction est placée dans un état d'attente. Une ligne est insérée dans la vue V\$SESSION_WAIT pour la session, avec la colonne EVENT contenant "statement suspended, wait error to be cleared".
- Une alerte d'opération suspendue est générée sur l'objet qui a besoin de ressources supplémentaires pour terminer l'instruction mise en suspens.

Terminer une instruction mise en suspens

Lorsque la situation ayant entraîné l'erreur disparaît (par exemple, suite à l'intervention du DBA ou une fois que l'espace de tri a été libéré par d'autres interrogations), l'exécution de l'instruction mise en suspens reprend automatiquement et l'alerte "resumable session suspended" est supprimée.

Il est possible de forcer une instruction mise en suspens à activer l'exception SERVERERROR en utilisant la procédure DBMS_RESUMABLE.ABORT().

Cette procédure peut être appelée par un DBA ou par l'utilisateur ayant exécuté l'instruction. Si le délai d'expiration de mise en suspens associé à l'instruction est atteint, l'instruction est automatiquement interrompue et une erreur est renvoyée à l'utilisateur.

Opérations pouvant faire l'objet d'une reprise

Les opérations suivantes peuvent faire l'objet d'une reprise :

- Interrogations : instructions `SELECT` qui manquent d'espace temporaire (pour les zones de tri)
- Commandes LMD : instructions `INSERT`, `UPDATE` et `DELETE`
- Instructions LDD suivantes :
 - `CREATE TABLE ... AS SELECT`
 - `CREATE INDEX`
 - `ALTER INDEX ... REBUILD`
 - `ALTER TABLE ... MOVE PARTITION`
 - `ALTER TABLE ... SPLIT PARTITION`
 - `ALTER INDEX ... REBUILD PARTITION`
 - `ALTER INDEX ... SPLIT PARTITION`
 - `CREATE MATERIALIZED VIEW`

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Opérations pouvant faire l'objet d'une reprise

Les opérations suivantes peuvent faire l'objet d'une reprise :

- **Interrogations** : Les instructions `SELECT` qui manquent d'espace temporaire (pour les zones de tri) peuvent être reprises. Avec OCI (Oracle Call Interface), les appels `OCISmtExecute()` et `OCISmtFetch()` sont candidats.
- **Commandes LMD** : Les instructions `INSERT`, `UPDATE` et `DELETE` sont candidates. Peu importe l'interface utilisée pour les exécuter : OCI, SQLJ, PL/SQL ou une autre interface. Les instructions `INSERT INTO ... SELECT` à partir de tables externes peuvent également être reprises.
- **Instructions LDD** : Les instructions suivantes peuvent faire l'objet d'une reprise :
 - `CREATE TABLE ... AS SELECT`
 - `CREATE INDEX`
 - `ALTER INDEX ... REBUILD`
 - `ALTER TABLE ... MOVE PARTITION`
 - `ALTER TABLE ... SPLIT PARTITION`
 - `ALTER INDEX ... REBUILD PARTITION`
 - `ALTER INDEX ... SPLIT PARTITION`
 - `CREATE MATERIALIZED VIEW`

Quiz

Sélectionnez les affirmations qui sont vraies concernant la fonctionnalité de gestion de l'espace :

1. Dans Oracle Database 11g Release 2, la création de segment est différée pour toutes les tables. Il n'y a pas d'exception.
2. Les index et les partitions d'index UNUSABLE sont créés sans segment.
3. La récupération d'espace dans les segments ne peut pas faire l'objet d'une reprise.
4. Vous pouvez définir des seuils par tablespace.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Réponses : 2, 4

Synthèse

Ce chapitre vous a permis d'apprendre à :

- décrire comment le serveur de base de données Oracle gère automatiquement l'espace
- économiser l'espace en utilisant la compression
- surveiller et gérer de façon proactive l'utilisation de l'espace dans les tablespaces
- décrire la création de segments dans une base Oracle
- utiliser la création différée de segments
- utiliser la fonction de conseil Segment Advisor
- récupérer l'espace inutilisé des tables et des index à l'aide de la fonctionnalité de récupération d'espace dans les segments
- gérer la reprise après un problème d'allocation d'espace

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Présentation de l'exercice 18 : Gérer le stockage

Cet exercice porte sur les points suivants :

- Utiliser des alertes avec seuil pour gérer les tablespaces de façon proactive
- Utiliser Segment Advisor pour effectuer une récupération d'espace
- Afficher les alertes et leur historique dans SQL*Plus et Enterprise Manager

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Unauthorized reproduction or distribution prohibited. Copyright© 2014, Oracle and/or its affiliates.

Laurent PERPETUE (laurent.perpetue@srr.fr) has a non-transferable license to use this Student Guide.

Gérer l'espace de la base de données

19

Copyright © 2009, Oracle. Tous droits réservés.

ORACLE

Laurent PERPETUE (laurent.perpetue@sir.fr) has a non-transferable license to use this Student Guide

Objectifs

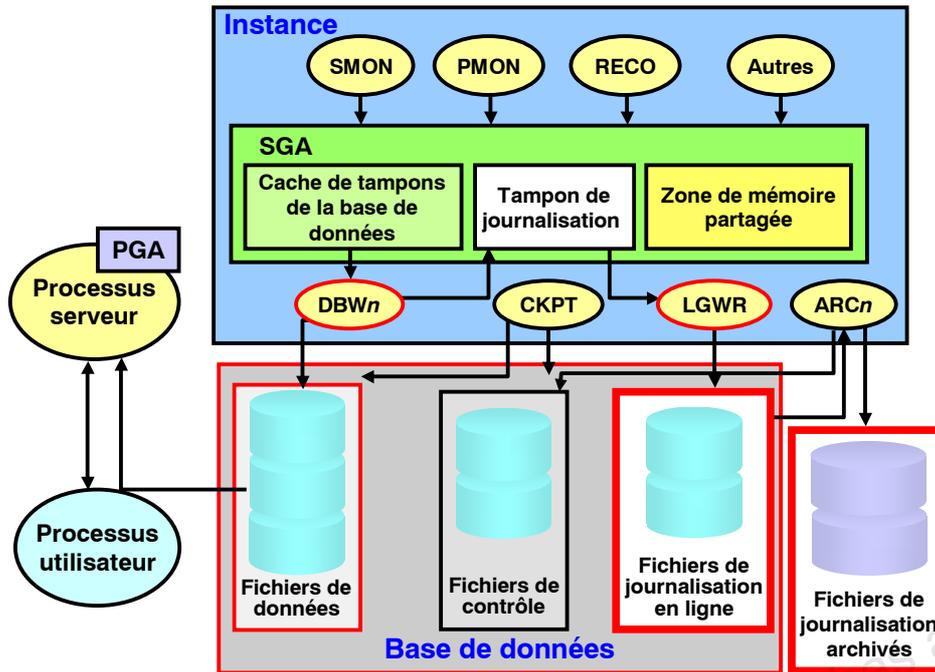
A la fin de ce chapitre, vous pourrez :

- décrire l'utilisation des disques avec secteurs de 4 ko
- utiliser les tablespaces transportables
- décrire les concepts relatifs aux tablespaces transportables

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Structures de stockage de la base de données



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Structures de stockage de la base de données

La base de données se compose de structures physiques et logiques. Celles-ci étant distinctes, il est possible de gérer le stockage physique des données sans affecter l'accès aux structures de stockage logiques.

Pour le stockage, la base de données utilise principalement des disques. De nos jours, ceux-ci ont généralement une taille de secteur de 512 ko, mais on commence à voir apparaître des disques avec secteurs de 4 ko, qui offrent une plus grande capacité de stockage et génèrent une surcharge moindre. Les bases Oracle accèdent à un disque dur via un pilote de périphérique spécifique à la plate-forme. Les processus Database Writer, Log Writer et ASM peuvent écrire directement sur le disque sans passer par le système d'exploitation (OS).

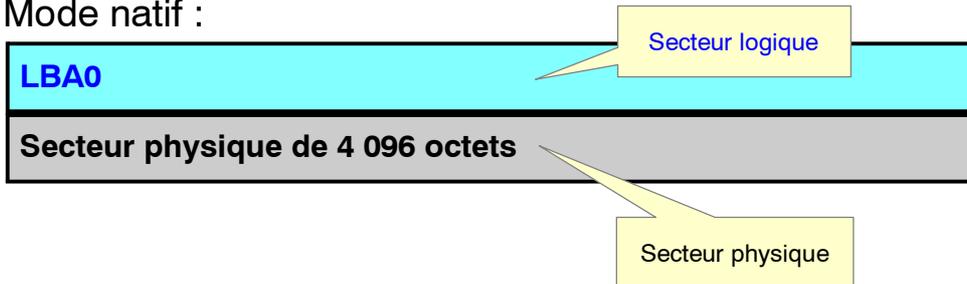
Oracle Database 11g Release 2 détecte la taille de secteur du disque. Il peut ainsi utiliser les disques de grande capacité sans dégradation des performances (grâce à des optimisations internes qui réduisent par exemple le gaspillage de l'espace d'annulation, lequel peut se produire avec les applications générant de nombreuses transactions courtes telles que les systèmes de messagerie).

Prise en charge des disques avec secteurs de 4 ko

- Mode émulation :



- Mode natif :



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Prise en charge des disques avec secteurs de 4 ko

Les disques avec secteurs de 4 ko possèdent des secteurs physiques (en gris) et des secteurs logiques (en bleu). Ces disques admettent deux modes, le mode émulation et le mode natif.

- En mode émulation, ils possèdent huit secteurs logiques par secteur physique (comme indiqué dans la diapositive). Ils présentent une interface de 512 octets avec leurs secteurs physiques de 4 ko. Cela signifie qu'une adresse LBA (Logical Block Address) référence 512 octets sur le disque.
Dans ce mode, les performances sont moins bonnes car l'unité de disque lit un secteur de 4 ko et le place dans le cache, change de section de 512 ko, puis réécrit l'ensemble du secteur de 4 ko sur le disque.
- En mode natif, les disques possèdent un secteur logique par secteur physique. Il n'y a donc que l'interface de 4 ko. En d'autres termes, une adresse LBA référence 4096 octets sur le disque.

Utiliser des disques avec secteurs de 4 ko

Mode émulation :

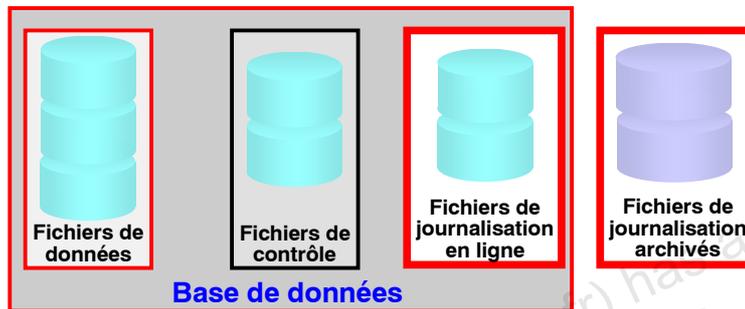
- Taille de bloc de 4 ko recommandée pour les fichiers de journalisation
- Taille de bloc de 4 ko (ou supérieure) recommandée pour les fichiers de données

Mode natif :

- Taille de bloc de 4 ko obligatoire pour les fichiers de journalisation
- Taille de bloc de 4 ko (ou supérieure) obligatoire pour les fichiers de données

Élément non affecté :

- Taille de bloc du fichier de contrôle : 16 ko



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Utiliser des disques avec secteurs de 4 ko

Dans Oracle Database 11gR2, les disques avec secteurs de 4 ko sont essentiellement utilisés pour les fichiers de journalisation (fichiers redo log). Ceux-ci incluent les fichiers de journalisation en ligne (online), les fichiers de journalisation de secours et les fichiers de journalisation archivés. Oracle recommande de créer des journaux dont la taille de bloc est de 4 ko sur les disques en mode émulation présentant des secteurs de 4 ko. Sur les disques en mode natif, vous devez créer des journaux dont la taille de bloc est de 4 ko.

En d'autres termes, la taille de bloc des fichiers de journalisation doit correspondre à celle des secteurs physiques du disque (pour les disques en mode natif avec secteurs de 512 octets et 4 ko). Si ce n'est pas le cas, vous recevez le message d'erreur ORA-1378. Pour les disques en mode émulation avec secteurs de 4 ko, la taille de bloc des fichiers de journalisation peut être de 512 ou 4 096 octets. La taille recommandée est de 4 ko. Lorsque vous créez des blocs de 512 octets sur des disques en mode émulation avec secteurs de 4 ko, un message d'avertissement est écrit dans le fichier d'alertes pour indiquer que la non-concordance des tailles de bloc entraîne une dégradation des performances. Cela s'applique également aux groupes de disques ASM.

Les disques avec secteurs de 4 ko ont également une incidence sur les fichiers de données Oracle Database. La base Oracle vous permet de créer des fichiers de données dont la taille de bloc est de 2 ko sur des disques avec secteurs de 512 octets. Sur des disques avec secteurs de 4 ko en mode émulation, Oracle recommande de créer des fichiers de données dont la taille de bloc est de 4 ko (ou plus). Sur les disques en mode natif avec secteurs de 4 ko, vous devez créer des journaux dont la taille de bloc est de 4 ko (ou plus).

La taille de bloc du fichier de contrôle est déjà de 16 ko. Elle n'est pas affectée par l'utilisation de disques avec secteurs de 4 ko.

Définir la taille de secteur d'un disque

A l'aide des clauses `SECTOR_SIZE` et `BLOCKSIZE` des commandes suivantes :

- `CREATE DISKGROUP`
- `ALTER DATABASE`
- `CREATE DATABASE`
- `CREATE CONTROL FILE`

Taille de secteur par défaut basée sur le matériel (et non l'ancienne taille de 512 octets)

```
CREATE DATABASE sample NORESETLOGS FORCE LOGGING
ARCHIVELOG
LOGFILE
  GROUP 1 '$ORACLE_BASE/oradata/sample/redo01.log'
          SIZE 100M BLOCKSIZE 4096,
  GROUP 2 '$ORACLE_BASE/oradata/sample/redo02.log'
          SIZE 100M BLOCKSIZE 4096
DATAFILE
...
```

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Définir la taille de secteur d'un disque

Dans un environnement ASM (Automatic Storage Management), il est possible de définir l'attribut `SECTOR_SIZE` pour des groupes de disques. Cette opération ne peut être effectuée que lors de la création du groupe à l'aide de la commande `CREATE DISKGROUP`.

Vous pouvez indiquer la taille du fichier de journalisation en utilisant la nouvelle clause `BLOCKSIZE` pour les commandes suivantes :

- `ALTER DATABASE`
- `CREATE DATABASE`
- `CREATE CONTROL FILE`

Aucune tâche supplémentaire n'est requise lorsque vous créez une base de données sur des disques présentant des secteurs de 4 ko, ce qui n'était pas le cas pour les disques avec secteurs de 512 octets. Aucune modification de l'environnement graphique n'est nécessaire.

Il vous est possible d'utiliser la clause `BLOCKSIZE` dans la commande `CREATE DATABASE` (voir la diapositive ci-dessus). Si vous ne définissez pas la taille de bloc, la base Oracle détecte la taille de secteur du disque sous-jacent et l'utilise pour déterminer la taille de bloc pour la création des fichiers de journalisation. Par défaut, celle-ci est la taille de secteur du disque, et non l'ancienne taille de 512 octets.

Quiz

Sur les disques en mode natif avec des secteurs de 4 ko, vous devez utiliser des fichiers journaux avec une taille de bloc de 4 ko.

1. Vrai
2. Faux

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Réponse : 1

Quiz

Oracle recommande de créer des blocs de 512 ko sur les disques en mode émulation présentant des secteurs de 4 ko.

1. Vrai
2. Faux

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Réponse : 2

Quiz

Les fichiers de contrôle ne sont pas affectés par l'utilisation de disques avec secteurs de 4 ko (car ils ont déjà une taille plus importante).

1. Vrai
2. Faux

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Réponse : 1

Transport de tablespaces

- Les tablespaces transportables entre plates-formes :
 - simplifient le déplacement des données entre un data warehouse et des datamarts
 - permettent la migration d'une base de données d'une plate-forme vers une autre
- Plates-formes prises en charge :

Solaris[tm] OE (32 bits)	HP-UX (64 bits)	Microsoft Windows IA (64 bits)
Solaris[tm] OE (64 bits)	HP Tru64 UNIX	Linux basé sur IBM zSeries
Microsoft Windows IA (32 bits)	HP-UX IA (64 bits)	Linux 64 bits pour AMD
Linux IA (32 bits)	Linux IA (64 bits)	Apple Mac OS
Systèmes basés sur AIX (64 bits)	HP Open VMS	Microsoft Windows 64 bits AMD
Linux basé sur IBM Power	HP IA Open VMS	Solaris x86 et AMD64

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Transport de tablespaces

Un tablespace transportable constitue le moyen le plus rapide de déplacer des volumes importants de données entre deux bases Oracle. A l'aide de tablespaces transportables, des fichiers de données Oracle (contenant des données de table, des index et quasiment tout objet de base de données Oracle) peuvent être directement transportés d'une base vers une autre. En outre, comme l'import et l'export, les tablespaces transportables fournissent un mécanisme permettant de transporter non seulement des données mais aussi des métadonnées.

Vous pouvez utiliser la fonctionnalité de tablespace transportable pour déplacer des données entre différentes plates-formes. Cela permet de simplifier la distribution des données entre un environnement de data warehouse et des datamarts, lesquels s'exécutent souvent sur des plates-formes plus restreintes. Cela permet également de migrer une base de données d'une plate-forme vers une autre, via la reconstruction du dictionnaire et le transport des tablespaces utilisateur.

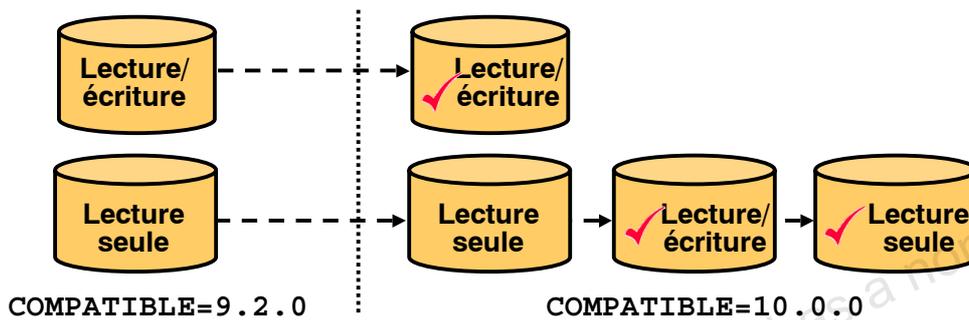
Le déplacement de données à l'aide de tablespaces transportables est bien plus rapide que la réalisation d'un export/import ou d'un déchargement/chargement des mêmes données. En effet, les fichiers contenant l'ensemble des données réelles sont juste copiés vers l'emplacement de destination et vous utilisez Data Pump pour transférer uniquement les métadonnées des objets de tablespace vers la nouvelle base.

Pour pouvoir transporter des fichiers de données d'une plate-forme vers une autre, vous devez vous assurer que le système source et le système cible s'exécutent sur l'une des plates-formes prises en charge (voir la diapositive ci-dessus).

Remarque : La fonctionnalité de tablespace transportable nécessite que la plate-forme source et la plate-forme cible utilisent le même jeu de caractères.

Concept : Niveau de compatibilité minimum

- Le paramètre COMPATIBLE doit avoir la valeur 10.0.0 ou une valeur supérieure pour les bases de données source et cible.
- La plate-forme est identifiée dans les en-têtes des fichiers de données (platform-aware).
- Avant le transport, assurez-vous que la plate-forme est identifiée dans les fichiers en lecture seule et dans les fichiers hors ligne.



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Concept : Niveau de compatibilité minimum

Pour permettre l'utilisation de la fonctionnalité de tablespace transportable entre plates-formes, il faut que le paramètre d'initialisation COMPATIBLE ait la valeur 10.0.0 (ou une valeur supérieure) pour les bases source et cible.

Lors de la première ouverture des fichiers de données sous Oracle Database 10g ou 11g avec une valeur de 10.0.0 (ou supérieure) pour le paramètre COMPATIBLE, les fichiers deviennent "platform-aware" (la plate-forme est identifiée dans les fichiers).

Cette caractéristique est représentée par les coches dans le schéma de la diapositive ci-dessus. Chaque fichier identifie la plate-forme à laquelle il appartient. Sur le disque, les fichiers présentent des formats identiques pour les blocs d'en-tête de fichier utilisés pour l'identification et la vérification des fichiers. La valeur du paramètre de compatibilité des fichiers en lecture seule et des fichiers hors ligne n'est incrémentée que lorsque ceux-ci sont placés en lecture/écriture ou lorsqu'ils sont mis en ligne. Par conséquent, les tablespaces qui sont en lecture seule dans des bases de données antérieures à Oracle Database 10g doivent être placés en lecture/écriture au moins une fois pour pouvoir utiliser la fonctionnalité de tablespace transportable entre plates-formes.

Niveau de compatibilité minimum

	Paramètre de compatibilité minimum	
	Base de données source	Base de données cible
Scénario de transport		
Bases de données sur la même plate-forme	8.0	8.0
Tablespace avec une taille de bloc de base de données différente de la base cible	9.0	9.0
Bases de données sur différentes plates-formes	10.0	10.0

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

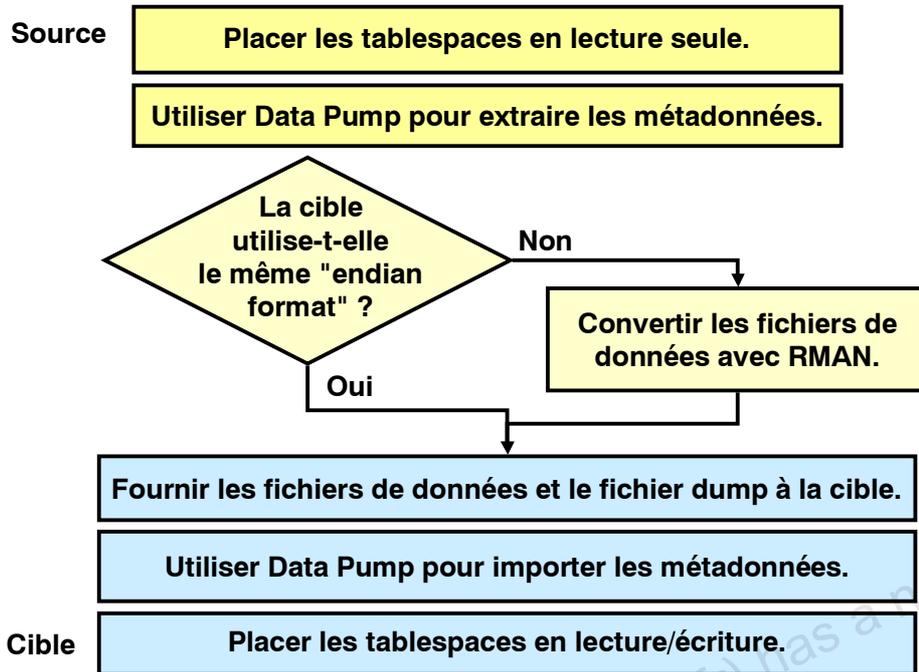
Niveau de compatibilité minimum

Lorsque vous créez un jeu de tablespaces transportables, Oracle Database calcule le niveau de compatibilité le plus bas auquel la base de données cible doit être exécutée. C'est ce qu'on appelle le niveau de compatibilité du jeu transportable. A compter d'Oracle Database 11g, un tablespace peut toujours être transporté vers une base de données avec un paramètre de compatibilité égal ou supérieur, et ce même si la base cible ne figure pas sur la même plate-forme. La base de données signale une erreur si le niveau de compatibilité du jeu transportable est supérieur à celui de la base cible.

Le tableau de la diapositive ci-dessus présente les exigences en matière de compatibilité minimale pour les tablespaces source et cible selon différents scénarios. Il n'est pas nécessaire que les bases de données source et cible présentent le même paramètre de compatibilité.

Lorsque les fichiers de données sont ouverts pour la première fois, chacun d'eux identifie la plate-forme à laquelle il appartient. Sur le disque, les fichiers présentent des formats identiques pour les blocs d'en-tête de fichier utilisés pour l'identification et la vérification des fichiers. La valeur du paramètre de compatibilité des fichiers en lecture seule et des fichiers hors ligne n'est incrémentée que lorsque ceux-ci sont placés en lecture/écriture ou lorsqu'ils sont mis en ligne.

Procédure relative aux tablespaces transportables



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Procédure relative aux tablespaces transportables

Pour pouvoir transporter un tablespace d'une plate-forme vers une autre (d'une plate-forme source vers une plate-forme cible), il faut d'abord convertir les fichiers de données appartenant au jeu de tablespaces dans un format pouvant être interprété par la base cible. Les structures de disque Oracle Database ont généralement un format commun, mais il est possible que les plates-formes source et cible utilisent des "endian formats" (formats de stockage et poids des données) différents (ordre des octets). Lorsque vous déplacez un tablespace vers une plate-forme présentant un "endian format" différent, vous devez utiliser la commande CONVERT de l'utilitaire RMAN pour modifier l'ordre des octets. Cette opération peut être effectuée sur la plate-forme source ou sur la plate-forme cible. Pour les plates-formes présentant le même "endian format", aucune conversion n'est requise.

Le schéma de la diapositive ci-dessus décrit les étapes possibles pour transporter un tablespace d'une plate-forme source vers une plate-forme cible. Il est aussi possible d'effectuer la conversion après avoir fourni les fichiers à la plate-forme cible. Les deux dernières étapes doivent être exécutées sur la plate-forme cible.

Pour l'essentiel, la procédure est la même qu'avec les versions antérieures du serveur de base de données Oracle, excepté lorsque les deux plates-formes utilisent des "endian formats" différents. On suppose que les deux plates-formes sont compatibles avec la fonctionnalité de tablespace transportable.

Remarque : L'ordre des octets peut affecter les résultats lors de l'écriture et de la lecture des données. Par exemple, la valeur entière 1 sur deux octets s'écrit 0x0001 sur un système avec l'octet de poids fort au début (big-endian), tel que Sun SPARC Solaris, tandis qu'elle s'écrit 0x0100 sur un système avec l'octet de poids faible à la fin (little-endian), tel qu'un PC compatible Intel.

Déterminer le "endian format" d'une plate-forme

```
SELECT tp.endian_format
FROM v$transportable_platform tp, v$database d
WHERE tp.platform_name = d.platform_name;
```

Source

Cible

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Déterminer le "endian format" d'une plate-forme

Vous pouvez interroger la vue V\$TRANSPORTABLE_PLATFORM pour déterminer si le "endian ordering" (ordre et poids des bits) est le même sur les deux plates-formes. La vue V\$DATABASE comporte deux colonnes permettant de déterminer le nom et l'identificateur de votre plate-forme. Exécutez l'interrogation ci-dessous pour obtenir la liste complète des plates-formes prises en charge et de leurs "endian formats" :

```
SQL> SELECT * FROM V$TRANSPORTABLE_PLATFORM;
```

PLATFORM_ID	PLATFORM_NAME	ENDIAN_FORMAT
1	Solaris[tm] OE (32-bit)	Big
2	Solaris[tm] OE (64-bit)	Big
7	Microsoft Windows IA (32-bit)	Little
10	Linux IA (32-bit)	Little
6	AIX-Based Systems (64-bit)	Big
3	HP-UX (64-bit)	Big
5	HP Tru64 UNIX	Little
4	HP-UX IA (64-bit)	Big
11	Linux IA (64-bit)	Little
15	HP Open VMS	Little

Déterminer le "endian format" d'une plate-forme (suite)

PLATFORM_ID	PLATFORM_NAME	ENDIAN_FORMAT
8	Microsoft Windows IA (64-bit)	Little
9	IBM zSeries Based Linux	Big
13	Linux 64-bit for AMD	Little
16	Apple Mac OS	Big
12	Microsoft Windows 64-bit for AMD	Little
17	Solaris Operating System (x86)	Little
18	IBM Power Based Linux	Big
19	HP IA Open VMS	Little
20	Solaris Operating System (AMD64)	Little

Utiliser la commande RMAN CONVERT

RMAN :

- convertit les tablespaces, les fichiers de données ou les bases au format d'une plate-forme de destination
- ne modifie pas les fichiers d'entrée
- écrit les fichiers convertis vers la destination

```
CONNECT TARGET SYS@orcl
RMAN>
SQL 'ALTER TABLESPACE hr READ ONLY';
CONVERT TABLESPACE hr
  TO PLATFORM 'Solaris[tm] OE (64-bit)'
  FORMAT '/tmp/transport_to_solaris/%U';;
```

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Utiliser la commande RMAN CONVERT

Vous utilisez la commande RMAN CONVERT pour convertir un tablespace, un fichier de données ou une base au format d'une plate-forme de destination afin de préparer le transport vers différentes plates-formes. Les fichiers d'entrée ne sont pas modifiés par la commande CONVERT car la conversion n'est pas effectuée "in place". RMAN écrit les fichiers convertis vers la destination indiquée.

Exemple utilisant CONVERT TABLESPACE :

- Supposons que vous disposiez d'une base ORCL sur une plate-forme Linux 32 bits et que vous vouliez la transporter sur une plate-forme Solaris 64 bits.
- Connectez-vous en tant que TARGET à la base de données *source* (montée ou ouverte).
- Le tablespace doit être en lecture seule au moment de la conversion.
- Le résultat est un ensemble de fichiers de données convertis dans le répertoire `/tmp/transport_to_solaris/`. Les données présentent le "endian format" approprié pour la plate-forme Solaris 64 bits.

Restrictions : La commande CONVERT ne traite pas les types de données utilisateur nécessitant des conversions de "endian format". Pour transporter entre différentes bases des objets qui sont construits sur des types sous-jacents stockant les données dans un format spécifique de la plate-forme, recourez aux utilitaires d'import et d'export Data Pump.

Pour connaître les prérequis, les modalités d'utilisation, les restrictions et la syntaxe, consultez le document *Oracle Database Backup and Recovery Reference*.

Tablespaces transportables avec Enterprise Manager

The screenshot displays the Oracle Enterprise Manager 11g interface for transporting tablespaces. It shows three main steps: 1. 'Transport Tablespaces' where 'Generate a transportable tablespace set' is selected and 'oracle' is chosen as the host credential. 2. 'Select Tablespaces' where 'EXAMPLE' and 'USERS' are selected from a list. 3. 'Checking Containment' where a progress indicator shows the containment check is being performed on the selected tablespaces. A red arrow points from the 'Check Containment' button in the second step to the 'Checking Containment' dialog box.

Copyright © 2009, Oracle. Tous droits réservés.

Tablespaces transportables avec Enterprise Manager

Enterprise Manager peut être utilisé pour implémenter des tablespaces transportables. Dans la page d'accueil de la base de données, cliquez sur l'onglet Data Movement, puis sur Transport Tablespaces dans la section Move Database Files. Sélectionnez "Generate a transportable table set" et fournissez les informations d'identification et de connexion (credentials) de l'utilisateur oracle, puis cliquez sur Continue. Dans la page Select Tablespaces, ajoutez les tablespaces à transporter depuis la liste affichée en cliquant sur le bouton Tablespace, puis sur Add. Vers le bas de la page, vous devez sélectionner le niveau de vérification de conteneur à appliquer avant le traitement des tablespaces. La vérification de conteneur recherche les dépendances d'objet au sein des tablespaces. Lorsque vous avez terminé, cliquez sur Next. Patientez pendant l'exécution de la vérification de conteneur. Réglez tous les problèmes détectés par la vérification avant de continuer.

Tablespaces transportables avec Enterprise Manager

The image displays two screenshots of the Oracle Enterprise Manager 11g interface, showing the 'Generate Transportable Tablespaces' wizard. The first screenshot is the 'Destination Characteristics' step (Step 2 of 6). It shows the source database 'ord' on 'Linux IA (32-bit)'. The 'Destination Database Platform' section has 'Linux IA (32-bit)' selected. The 'Destination Character Set' section has 'AL32UTF8' and 'AL16UTF16' selected. The 'Next' button is highlighted. A red arrow points from the 'Next' button to the second screenshot. The second screenshot is the 'Files' step (Step 4 of 6). It shows the 'Dump File' section with 'Dump File Directory' set to '/u01/app/oracle/oradata/ord/' and 'Dump File Name' set to 'EXPDAT_GENERATETTS000041.DMP'. The 'Datfiles' section has 'Copy Datfiles Automatically (Recommended)' selected. The 'Next' button is highlighted.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Tablespaces transportables avec Enterprise Manager (suite)

Dans la page Destination Characteristics, vous devez indiquer la plate-forme de destination et les jeux de caractères. Dans la section Destination Database Platform, sélectionnez dans la liste déroulante le système d'exploitation de l'ordinateur de destination. Si la plate-forme de destination est différente de la plate-forme source, Enterprise Manager effectue une conversion de données. Accédez ensuite à la section Destination Character Set et choisissez dans les listes déroulantes le jeu de caractères de destination ainsi que le jeu de caractères nationaux. Ces jeux de caractères doivent être compatibles avec les jeux source. Lorsque vous cliquez sur Next pour continuer, Enterprise Manager vérifie la compatibilité des jeux de caractères. Si ceux choisis sont signalés comme incompatibles, vous revenez à la page Destination Characteristics pour corriger vos sélections.

Tablespaces transportables avec Enterprise Manager

Generate Transportable Tablespaces: Schedule

Database: **orcl**
Logged In As: **SYS**

Specify a name and description for the export job. Specify a date to start the job.

Job Parameters

Job Name: **GENERATETTS000041**
Description: **Transport Tablespace(s)**

Job Schedule

Start

Immediately

Generate Transportable Tablespaces: Review

Database: **orcl**
Logged In As: **SYS**

Summary

Job Name: **GENERATETTS000041**
Transport Tablespaces: **EXAMPLE,USERS**
Containment Type: **SELF**
Include Constraints: **false**
Source Platform: **Linux IA (32-bit)**
Destination Platform: **Linux IA (32-bit)**
Disk Space Required (MB): **105**
Copy the datafiles: **AUTO**
Dump File Format: **Datapump**

Dump File

Dump File: **/u01/app/oracle/oradata/orcl/**
Directory:
Dump File Name: **EXPDAT_GENERATETTS000041.DMP**

Datafiles

Datafile Name	New Datafile Name	Size (MB)
/u01/app/oracle/oradata/orcl/users01.dbf	/u01/app/oracle/oradata/orcl/users01.dbf	5
/u01/app/oracle/oradata/orcl/example01.dbf	/u01/app/oracle/oradata/orcl/example01.dbf	100

Copyright © 2009, Oracle. Tous droits réservés.

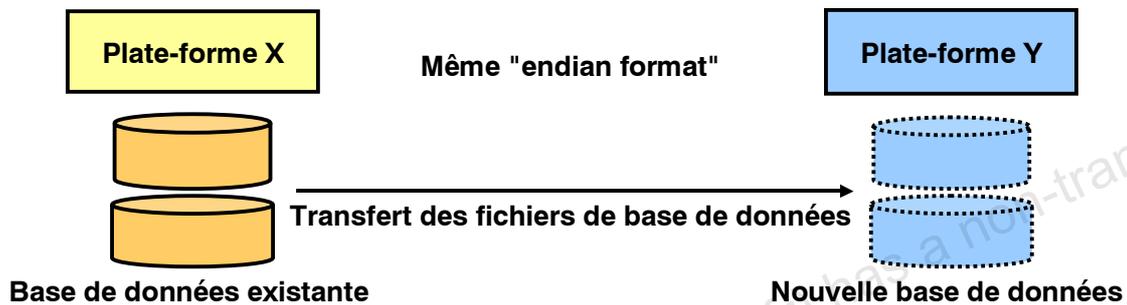
Tablespaces transportables avec Enterprise Manager (suite)

Dans la page Schedule, fournissez une description significative pour le nom de travail par défaut. Vous pouvez également choisir de démarrer le travail immédiatement ou de planifier une exécution ultérieure. Une fois vos sélections effectuées, cliquez sur Next pour continuer. Dans la page Review, vous pouvez vérifier vos choix avant de soumettre le travail pour exécution. Si les entrées sont correctes, cliquez sur le bouton Submit Job.

Pour corriger des entrées incorrectes, cliquez sur le bouton Back.

Transport de base de données

- Il généralise la fonctionnalité de tablespace transportable.
- Les sous-ensembles de données peuvent être facilement distribués d'un environnement de data warehouse vers des datamarts, lesquels s'exécutent souvent sur des plates-formes plus restreintes.
- Une base de données peut être migrée d'une plate-forme à une autre très rapidement.



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Transport de base de données

Vous pouvez utiliser la fonctionnalité de tablespace transportable pour migrer une base de données vers une autre plate-forme : vous créez une nouvelle base sur la plate-forme de destination, puis procédez au transport de tous les tablespaces utilisateur. Toutefois, vous ne pouvez pas transporter le tablespace `SYSTEM`. Par conséquent, les objets tels que les séquences, les packages PL/SQL et d'autres objets qui dépendent du tablespace `SYSTEM` ne peuvent pas être transportés de cette manière. Vous devez créer ces objets manuellement dans la base de données de destination ou utiliser Data Pump pour les transporter.

Pour pouvoir transporter des bases de données d'une plate-forme à une autre, vous devez vous assurer que le système source et le système cible sont exécutés sur l'une des plates-formes répertoriées dans la vue `V$TRANSPORTABLE_PLATFORM` et que les deux systèmes ont le même "endian format". Par exemple, vous pouvez transporter une base de données exécutée sur Linux IA (32 bits) vers l'une des plates-formes Windows.

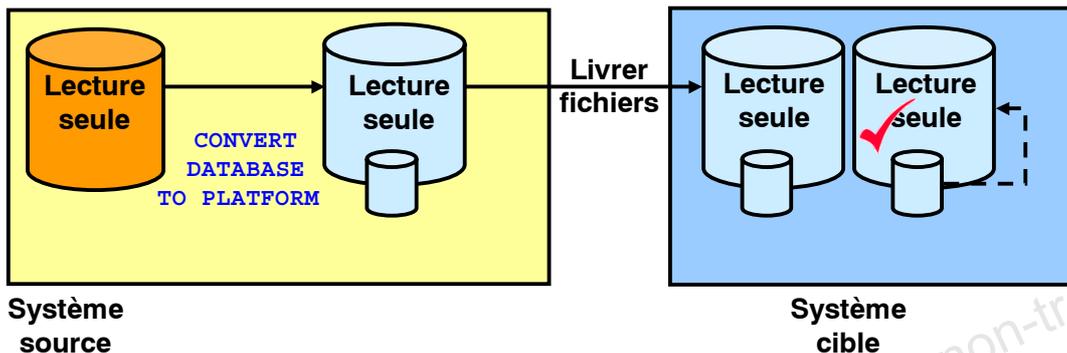
Si l'une des bases de données ou les deux utilisent Automatic Storage Management (ASM), vous pouvez être amené à utiliser le package `DBMS_FILE_TRANSFER` pour transférer les fichiers.

Dans le cadre du transport de tablespace, une base de données cible est déjà disponible pour l'ajout des données. La fonctionnalité de transport de base de données crée quant à elle une nouvelle base sur la plate-forme cible. La base créée contient les mêmes données que la base source. Elle possède également les mêmes propriétés que la base source, à l'exception du nom de base de données, du nom d'instance et de l'emplacement des fichiers.

Remarque : Le transport d'une base de données est plus rapide que l'utilisation de Data Pump pour déplacer les données.

Procédure de transport de base de données : Conversion du système source

Ouvrir la base de données en mode **READ ONLY**
avec **COMPATIBLE=10.0.0** ou une valeur supérieure.



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Procédure de transport de base de données : Conversion du système source

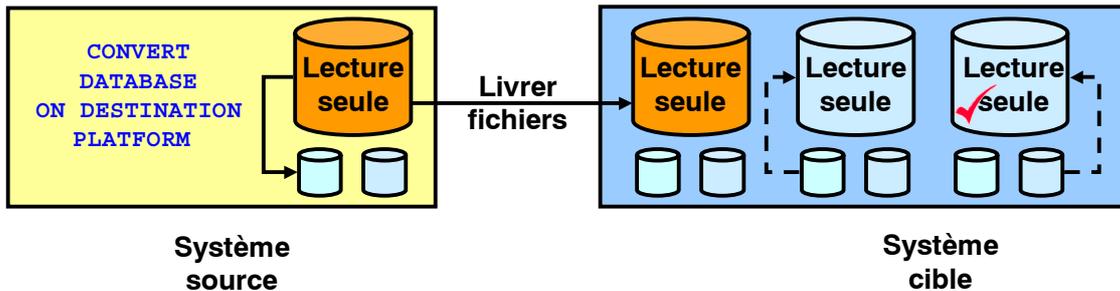
Pour pouvoir transporter une base de données, vous devez l'ouvrir en mode **READ ONLY**. Utilisez ensuite **RMAN** pour convertir les fichiers de données requis de la base.

Lorsque vous effectuez la conversion sur la plate-forme source, la commande **RMAN CONVERT DATABASE** génère un script qui contient la commande **CREATE CONTROLFILE RESETLOGS** appropriée, utilisée sur le système cible pour créer la nouvelle base. La commande **CONVERT DATABASE** convertit ensuite tous les fichiers de données identifiés pour qu'ils puissent être utilisés sur le système cible. Vous transférez alors vers la plate-forme cible les fichiers de données convertis et le script généré. En exécutant le script généré sur la plate-forme cible, vous créez une copie de la base.

Remarque : La base de données source doit être exécutée avec un paramètre d'initialisation **COMPATIBLE** qui a la valeur **10.0.0** ou une valeur supérieure. Tous les tablespaces identifiés doivent avoir été passés en mode **READ WRITE** au moins une fois après l'attribution de la valeur **10.0.0** (ou d'une valeur supérieure) au paramètre **COMPATIBLE**.

Procédure de transport de base de données : Conversion du système cible

Ouvrir la base de données en mode **READ ONLY**
avec **COMPATIBLE=10.0.0** ou une valeur supérieure.



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Procédure de transport de base de données : Conversion du système cible

Pour pouvoir transporter une base de données, vous devez l'ouvrir en mode **READ ONLY**. Utilisez ensuite **RMAN** pour convertir les fichiers de données requis de la base.

Si vous effectuez la conversion sur la plate-forme cible, la commande **CONVERT DATABASE** (qui est exécutée sur le système source) génère uniquement deux scripts qui sont utilisés sur le système cible pour convertir les fichiers de données et pour recréer les fichiers de contrôle de la nouvelle base. Vous transférez ensuite vers la plate-forme cible les fichiers de données identifiés et les deux scripts. Une fois cette opération terminée, exécutez les deux scripts dans l'ordre approprié. Le premier script utilise la commande **RMAN CONVERT DATAFILE** existante pour effectuer la conversion. Le second exécute la commande **SQL CREATE CONTROLFILE RESETLOGS** avec les fichiers de données convertis pour créer la nouvelle base.

Remarque : La base de données source doit être exécutée avec un paramètre d'initialisation **COMPATIBLE** qui a la valeur **10.0.0** ou une valeur supérieure. Tous les tablespaces identifiés doivent avoir été passés en mode **READ WRITE** au moins une fois après l'attribution de la valeur **10.0.0** (ou d'une valeur supérieure) au paramètre **COMPATIBLE**.

Transport d'une base de données : Considérations

- Créez le fichier de mots de passe sur la plate-forme cible.
- Transportez les fichiers `BFILE` utilisés dans la base de données source.
- Le fichier `PFILE` généré et le script de transport utilisent `OMF`.
- Utilisez `DBNEWID` pour modifier le `DBID`.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Transport d'une base de données : Considérations

Les fichiers de journalisation (redo logs), les fichiers de contrôle et les fichiers temporaires ne sont pas transportés. Ils sont recréés pour la nouvelle base de données, sur la plate-forme cible. Par conséquent, la nouvelle base de la plate-forme cible doit être ouverte avec l'option `RESETLOGS`.

Si un fichier de mots de passe est utilisé, il n'est pas transporté et vous devez le créer sur la plate-forme cible. En effet, les types de nom de fichier autorisés pour le fichier de mots de passe sont propres au système d'exploitation. Cependant, le résultat de la commande `CONVERT DATABASE` répertorie tous les noms utilisateur et les privilèges système associés. Il conseille de recréer le fichier de mots de passe et d'ajouter des entrées pour ces utilisateurs sur la plate-forme cible.

La commande `CONVERT DATABASE` permet de répertorier tous les objets répertoire et tous les objets qui utilisent des types de données `BFILE` ou des tables externes dans la base source. Vous pouvez être amené à mettre à jour ces objets avec les nouveaux noms de répertoire et de fichier. Si des fichiers `BFILE` sont utilisés dans la base de données, vous devez les transporter.

Le fichier `PFILE` généré et le script de transport utilisent Oracle Managed Files (OMF) pour les fichiers de base de données. Si vous ne souhaitez pas utiliser OMF, vous devez modifier le fichier `PFILE` et le script de transport.

La base de données transportée a le même `DBID` que la base source. L'utilitaire `DBNEWID` permet de modifier le `DBID`. Dans le script de transport et dans le résultat de la commande `CONVERT DATABASE`, vous êtes invité à vous servir de l'utilitaire `DBNEWID` pour modifier l'`ID` de la base de données.

Quiz

Sélectionnez les affirmations qui sont vraies :

1. La commande RMAN CONVERT effectue une conversion "in-place", de sorte que les fichiers d'entrée sont modifiés avant d'être transportés vers la destination.
2. Les tablespaces en lecture/écriture doivent être en lecture seule au moment de la conversion de "endian format".
3. Vous pouvez utiliser la commande RMAN CONVERT pour les tables, les tablespaces et les bases de données.
4. Vous pouvez transporter des bases de données dans un environnement de data warehouse.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Réponses : 2, 4

Synthèse

Ce chapitre vous a permis d'apprendre à :

- décrire l'utilisation des disques avec secteurs de 4 ko
- décrire les concepts relatifs aux tablespaces et aux bases de données transportables

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Présentation de l'exercice 19 : Gérer l'espace de la base de données

Cet exercice porte sur le point suivant :

- Visualiser une démonstration sur l'utilisation des disques avec secteurs de 4 ko

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

20

Dupliquer une base de données

Laurent PERPETUE (laurent.perpetue@sir.fr) has a non-transferable license to use this Student Guide

Objectifs

A la fin de ce chapitre, vous pourrez :

- énumérer les motifs de création d'une base de données dupliquée
- choisir une technique de duplication
- dupliquer une base de données à l'aide de RMAN
- utiliser une sauvegarde RMAN pour dupliquer une base de données
- dupliquer une base de données à partir d'une instance en cours d'exécution

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Utiliser une base de données dupliquée

- Utiliser une base de données dupliquée pour :
 - tester les procédures de sauvegarde et de récupération
 - récupérer des objets en créant un export, puis en important les objets dans la base de données de production
- Créer une base de données dupliquée :
 - A l'aide de la commande RMAN `DUPLICATE`
 - Sur le même hôte ou sur des hôtes distincts
 - Avec un contenu identique ou un sous-ensemble de la source
 - Via des canaux auxiliaires pour la duplication à partir de sauvegardes
 - Via des canaux cibles pour la duplication d'une base de données active

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Utiliser une base de données dupliquée

Une base de données dupliquée est une copie d'une base source à laquelle est attribuée un nouvel ID unique (DBID). Vous pouvez l'utiliser indépendamment de cette dernière pour les opérations suivantes :

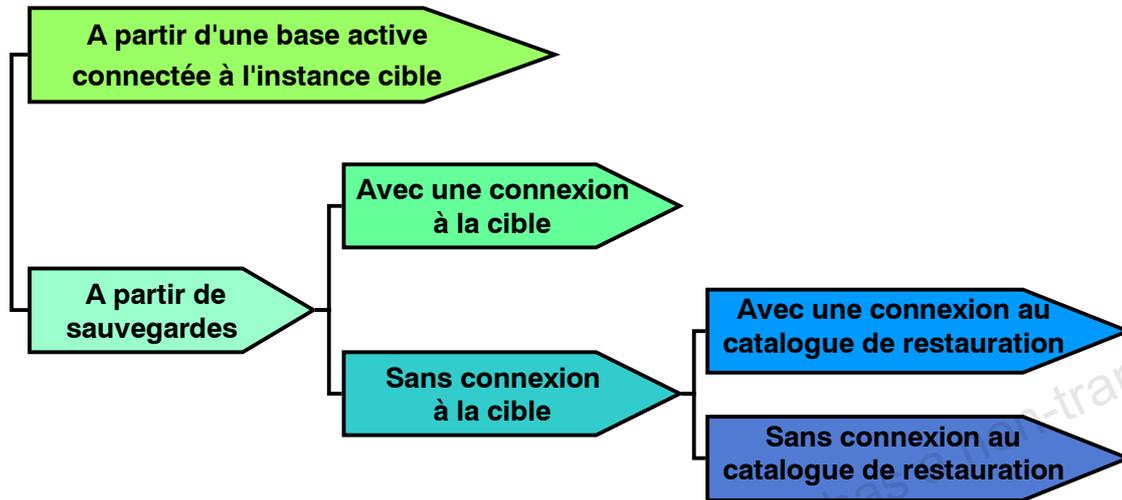
- Tester les procédures de sauvegarde et de récupération.
- Récupérer des objets qui ont été supprimés par inadvertance de la base cible, en créant un export contenant les objets de la base de données dupliquée et en effectuant un import dans la base de production. Toutefois, Flashback Query, Flashback Drop et Flashback Table constituent des solutions plus simples et plus rapides pour récupérer des objets.

Créer une base de données dupliquée :

- Vous pouvez utiliser la commande RMAN `DUPLICATE` pour créer une base de données dupliquée sur le même hôte ou sur des hôtes distincts.
- La base dupliquée peut avoir le même contenu que la base source, ou seulement un sous-ensemble (ce point sera détaillée plus loin dans le chapitre).
- La plus grande partie du travail de duplication est effectuée par les canaux auxiliaires. Pour la duplication à partir de sauvegardes, ces canaux correspondent à une session serveur sur l'instance auxiliaire de l'hôte cible.
- Pour la duplication d'une base de données active, ce sont les canaux cibles qui assurent la transmission des copies des fichiers de données vers l'instance auxiliaire.

Choisir la technique de duplication

Choix de la technique à utiliser pour dupliquer une base — toujours avec une connexion à l'instance auxiliaire :



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

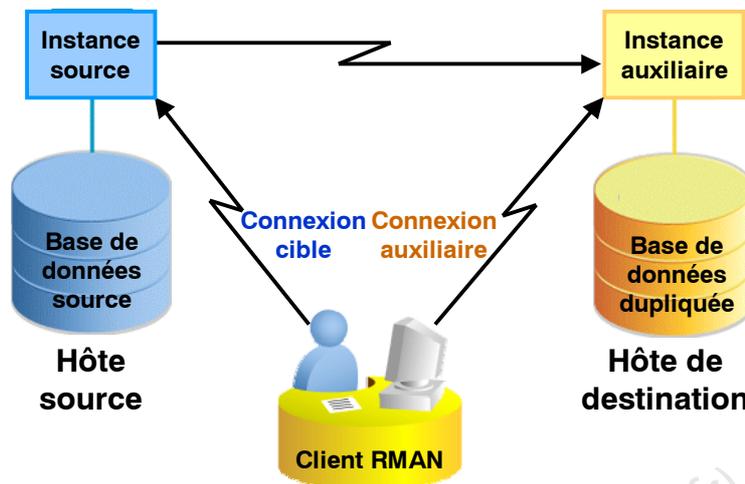
Choisir la technique de duplication

Vous pouvez dupliquer une base de données source pour obtenir une base cible située sur le même ordinateur ou sur une autre machine. L'instance de base de données associée à la base dupliquée est appelée l'instance auxiliaire. Toutes les techniques de duplication nécessitent une connexion à cette instance. Le diagramme de la diapositive présente les techniques de duplication suivantes :

- A partir d'une base de données active, avec une connexion à l'instance cible et à l'instance auxiliaire.
- A partir d'une sauvegarde, avec une connexion à l'instance cible et à l'instance auxiliaire.
- A partir d'une sauvegarde, avec une connexion à l'instance auxiliaire mais pas à l'instance cible, et avec une connexion au catalogue de restauration.
- A partir d'une sauvegarde, avec une connexion à l'instance auxiliaire mais pas à l'instance cible ni au catalogue de restauration.

Dupliquer une base de données active

- Avec le réseau (aucune sauvegarde n'est requise)
- Avec un fichier `SPFILE` personnalisé
- Via Enterprise Manager ou la ligne de commande RMAN



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

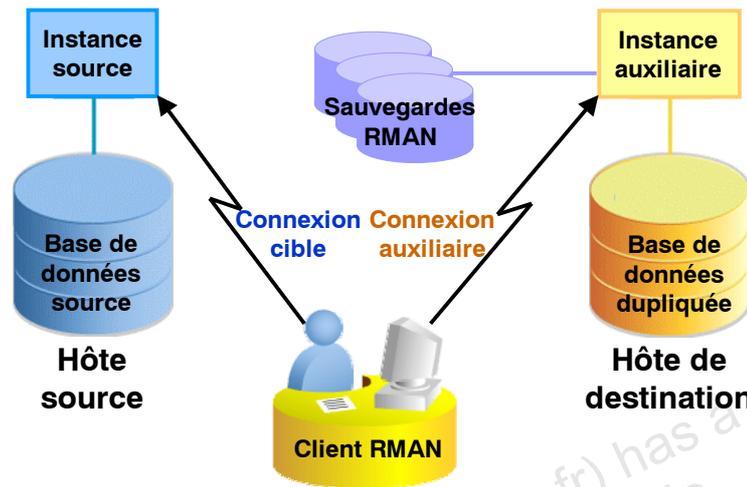
Dupliquer une base de données active

Vous pouvez ordonner à la base de données source d'effectuer des copies d'image en ligne et des copies de fichier de journalisation archivé (archived redo log) directement vers l'instance auxiliaire à l'aide d'Enterprise Manager ou de la clause `FROM ACTIVE DATABASE` de la commande `DUPLICATE` de RMAN. Il n'y a pas besoin de sauvegardes pour cette opération. RMAN se connecte en tant que `TARGET` à l'instance de base de données source et en tant qu'`AUXILIARY` à l'instance auxiliaire (comme indiqué dans la diapositive).

Les fichiers de base de données sont copiés de la source vers la destination (ou instance `AUXILIARY`) via une connexion réseau interinstance. RMAN utilise ensuite un "script mémoire" (c'est-à-dire présent uniquement en mémoire) pour effectuer la récupération des données et ouvrir la base.

Dupliquer une base de données avec une connexion à la cible

- Connectez-vous à la cible (base de données source).
- Connectez-vous à l'instance auxiliaire.
- Vous pouvez aussi vous connecter au catalogue de restauration (ou utiliser le fichier de contrôle de la cible).



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

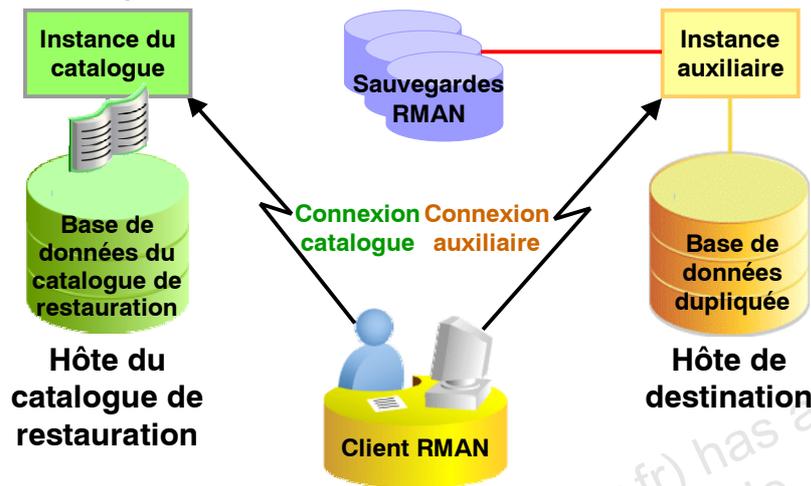
Dupliquer une base de données avec une connexion à la cible

Lorsque vous dupliquez une base de données avec une connexion à la base de données cible, RMAN peut obtenir les métadonnées relatives aux sauvegardes à partir du fichier de contrôle de la base de données cible ou à partir du catalogue de restauration.

Le diagramme de la diapositive illustre la duplication à partir de sauvegardes avec une connexion à la cible. RMAN se connecte à l'instance de base de données source et à l'instance auxiliaire. RMAN peut aussi se connecter à la base de données du catalogue de restauration (non représentée dans le diagramme). L'hôte de destination doit avoir accès aux sauvegardes RMAN qui sont nécessaires pour créer la base dupliquée.

Dupliquer une base de données à l'aide du catalogue de restauration sans connexion à la cible

- Connectez-vous au catalogue de restauration pour obtenir les métadonnées relatives aux sauvegardes.
- Connectez-vous à l'instance auxiliaire, qui doit avoir accès aux sauvegardes RMAN.



Copyright © 2009, Oracle. Tous droits réservés.

ORACLE

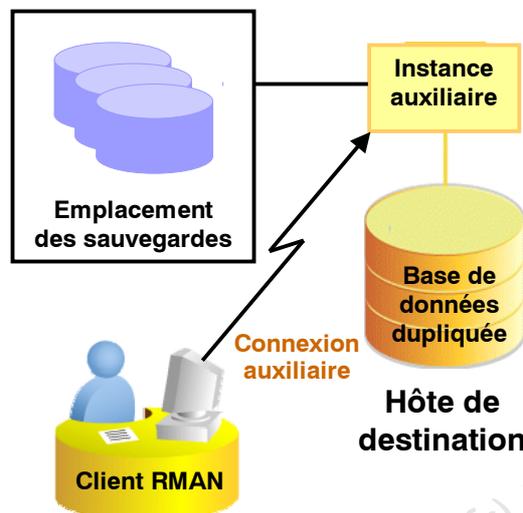
Dupliquer une base de données à l'aide du catalogue de restauration sans connexion à la cible

Lorsque vous dupliquez une base de données sans connexion à celle-ci mais avec une connexion au catalogue de restauration, RMAN utilise ce dernier pour obtenir les métadonnées relatives aux sauvegardes.

Le diagramme de la diapositive illustre la duplication à partir de sauvegardes sans connexion à la cible. RMAN se connecte à l'instance de base de données du catalogue de restauration et à l'instance auxiliaire. L'hôte de destination doit avoir accès aux sauvegardes RMAN qui sont nécessaires pour créer la base dupliquée.

Dupliquer une base de données sans catalogue de restauration ni connexion à la cible

Connectez-vous à l'instance auxiliaire, qui doit avoir accès à l'emplacement des sauvegardes sur disque.



ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Dupliquer une base de données sans catalogue de restauration ni connexion à la cible

Lorsque vous dupliquez une base de données sans connexion à celle-ci et sans connexion au catalogue de restauration, RMAN utilise l'emplacement où sont stockées les sauvegardes et les copies nécessaires (BACKUP LOCATION).

Le diagramme de la diapositive illustre la duplication à partir de sauvegardes sans connexion à la cible ni au catalogue de restauration. Sur l'hôte de destination, vous devez disposer d'un emplacement où sont stockées les sauvegardes ou les copies nécessaires à la duplication.

Créer une base de données dupliquée à partir de sauvegardes

1. Créez un fichier de mots de passe Oracle pour l'instance auxiliaire.
2. Etablissez la connectivité Oracle Net à l'instance auxiliaire.
3. Créez un fichier de paramètres d'initialisation pour l'instance auxiliaire
4. Démarrez l'instance auxiliaire en mode NOMOUNT.
5. Montez ou ouvrez la base de données cible.
6. Vérifiez la disponibilité des sauvegardes et des fichiers de journalisation archivés.
7. Si nécessaire, allouez des canaux auxiliaires.
8. Exécutez la commande DUPLICATE.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Créer une base de données dupliquée à partir de sauvegardes

Il est important de bien comprendre les étapes élémentaires décrites ci-après ainsi que le processus RMAN de duplication de base de données.

Si vous utilisez l'interface Enterprise Manager, la plupart de ces étapes sont réalisées par les assistants. En revanche, si vous créez la base dupliquée à partir de la ligne de commande, vous devez effectuer les différentes opérations manuellement. Vous pouvez utiliser l'interface EM comme outil de modélisation et créer votre propre processus de duplication de base de données à partir du journal généré.

Les principales étapes permettant de créer une base de données dupliquée sont présentées sur la diapositive. Certaines de ces étapes sont décrites plus en détail dans la suite du chapitre.

Créer un fichier de paramètres d'initialisation pour l'instance auxiliaire

Indiquez les paramètres suivants :

- **DB_NAME**
 - Si la base de données dupliquée figure dans le même répertoire d'origine Oracle Home que la base cible, leurs noms doivent être différents.
 - Utilisez la même valeur dans la commande `DUPLICATE`.
- **DB_BLOCK_SIZE**
 - Indiquez la même valeur que celle définie pour la base de données cible.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Créer un fichier de paramètres d'initialisation pour l'instance auxiliaire

Vous devez créer un fichier texte de paramètres d'initialisation pour l'instance auxiliaire. Ce fichier doit résider sur le même hôte que le client RMAN utilisé pour exécuter la commande `DUPLICATE`.

Tenez compte des exigences concernant chacun des paramètres suivants :

- **DB_NAME** : Si la base de données cible et la base dupliquée figurent dans le même répertoire d'origine Oracle Home, vous devez indiquer un nom distinct pour `DB_NAME`. Si elles figurent dans des répertoires d'origine Oracle Home différents, vous devez vérifier que le nom de la base dupliquée ne correspond pas à un autre nom existant dans le même répertoire. Lors de l'exécution de la commande `DUPLICATE`, veillez à utiliser le même nom de base de données que celui défini pour ce paramètre.
- **DB_BLOCK_SIZE** : La taille de bloc de la base de données auxiliaire doit correspondre à la taille de bloc de la base cible. Indiquez dans le fichier de paramètres d'initialisation de la base auxiliaire la même valeur que celle définie dans le fichier de paramètres d'initialisation de la base cible. Si le paramètre n'est pas défini dans le fichier de paramètres d'initialisation de la base cible, ne le définissez pas dans celui de l'instance auxiliaire.

Veillez également à vérifier la configuration de tous les paramètres d'initialisation indiquant des chemins. Vérifiez que tous les chemins indiqués sont accessibles sur l'hôte de la base de données dupliquée.

Indiquer de nouveaux noms pour la destination

Techniques disponibles :

- Commande `SET NEWNAME`
- Commande `CONFIGURE AUXNAME` (commande en phase d'abandon pour les fichiers de données du jeu de récupération)
- Paramètre `DB_FILE_NAME_CONVERT` avec la commande `DUPLICATE`

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Indiquer de nouveaux noms pour la destination

Vous pouvez utiliser les techniques suivantes pour indiquer de nouveaux noms pour les fichiers de données :

- Incluez la commande `SET NEWNAME FOR DATAFILE` au sein d'un bloc `RUN` pour indiquer de nouveaux noms pour les fichiers de données.
- Utilisez la commande `CONFIGURE AUXNAME`.
`CONFIGURE AUXNAME` est une alternative à `SET NEWNAME`. La différence entre les deux est que lorsque le nom auxiliaire est configuré, les commandes `DUPLICATE` suivantes réutilisent les paramètres configurés. En revanche, vous devez exécuter la commande `SET NEWNAME` chaque fois que vous lancez une duplication.
Remarque : `SET NEWNAME` remplace `CONFIGURE AUXNAME` pour les fichiers de données du jeu de récupération.
- Indiquez le paramètre `DB_FILE_NAME_CONVERT` avec la commande `DUPLICATE`.

Utiliser les clauses SET NEWNAME

- Les clauses SET NEWNAME permettent d'indiquer un format de nom par défaut pour tous les fichiers d'une base de données ou d'un tablespace nommé.
- Le nom par défaut est utilisé pour les commandes DUPLICATE, RESTORE et SWITCH du bloc RUN.
- Vous pouvez ainsi définir les noms de fichier en une seule commande au lieu de les indiquer individuellement.

```
SET NEWNAME FOR DATABASE  
TO {NEW| 'formatSpec'};
```

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Utiliser les clauses SET NEWNAME

Vous pouvez utiliser SET NEWNAME pour définir le format de nom par défaut de tous les fichiers de données d'un tablespace nommé et tous les fichiers de données de la base.

L'ordre de priorité associé à la commande SET NEWNAME est le suivant :

1. SET NEWNAME FOR DATAFILE et SET NEWNAME FOR TEMPFILE
2. SET NEWNAME FOR TABLESPACE
3. SET NEWNAME FOR DATABASE

Exemple :

```
RUN  
{  
SET NEWNAME FOR DATABASE TO  
  '/u01/app/oracle/oradata/dupldb/%b';  
DUPLICATE TARGET DATABASE TO dupldb  
LOGFILE  
GROUP 1 ('/u01/app/oracle/oradata/dupldb/redo01a.log',  
  '/u01/app/oracle/oradata/dupldb/redo01b.log') SIZE 50M REUSE,  
GROUP 2 ('/u01/app/oracle/oradata/dupldb/redo02a.log',  
  '/u01/app/oracle/oradata/dupldb/redo02b.log') SIZE 50M REUSE,  
GROUP 3 ('/u01/app/oracle/oradata/dupldb/redo03a.log',  
  '/u01/app/oracle/oradata/dupldb/redo03b.log') SIZE 50M REUSE;  
}
```

Variables de substitution pour SET NEWNAME

Élément de syntaxe	Description
%b	Définit le nom de fichier sans le chemin d'accès
%f	Définit le numéro absolu du fichier de données pour lequel le nouveau nom est généré
%I	Définit l'ID de BdD
%N	Définit le nom du tablespace
%U	Définit un nom de fichier généré par le système au format suivant : data-D-%d_id-%I_TS-%N_FNO-%f

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Variables de substitution pour SET NEWNAME

Lorsque vous générez SET NEWNAME FOR DATABASE ou SET NEWNAME FOR TABLESPACE, vous devez définir des variables de substitution dans la clause TO <nomdefichier> pour éviter tout conflit de nom. Indiquez au moins l'une des variables de substitution suivantes : %b, %f et %U. Les variables %I et %N sont facultatives.

Définir des paramètres pour les noms de fichier

Vous pouvez indiquer les paramètres suivants pour contrôler explicitement les noms des fichiers dans la base de données auxiliaire :

- CONTROL_FILES
- DB_FILE_NAME_CONVERT
- LOG_FILE_NAME_CONVERT

```
CONTROL_FILES='/u01/app/oracle/oradata/aux/control01.ctl',
              '/u01/app/oracle/oradata/aux/control02.ctl',
              '/u01/app/oracle/oradata/aux/control03.ctl'
DB_FILE_NAME_CONVERT='/u01/app/oracle/oradata/orcl',
                    '/u01/app/oracle/oradata/aux'
LOG_FILE_NAME_CONVERT='/u01/app/oracle/oradata/orcl',
                    '/u01/app/oracle/oradata/aux'
```

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Définir des paramètres pour les noms de fichier

RMAN génère des noms pour les fichiers de base de données requis lors de l'exécution de la commande DUPLICATE. Vous pouvez contrôler ces noms en indiquant les paramètres suivants dans le fichier de paramètres d'initialisation de l'instance auxiliaire :

- **CONTROL_FILES** : Associez à ce paramètre le nom des fichiers de contrôle. Si vous ne définissez pas les noms via ce paramètre, le serveur Oracle crée un fichier de contrôle géré par Oracle vers une destination par défaut. Pour obtenir des informations spécifiques, reportez-vous à la section relative à la commande SQL CREATE CONTROLFILE dans le manuel SQL Reference.
- **DB_FILE_NAME_CONVERT** : Ce paramètre permet d'indiquer le nom des fichiers de données pour la base auxiliaire. Il présente le format DB_FILE_NAME_CONVERT = '*string1*', '*string2*', où *string1* est le modèle pour les noms de fichier dans la base cible et *string2* est celui pour la base auxiliaire. Vous pouvez aussi indiquer le paramètre DB_FILE_NAME_CONVERT en option dans la commande DUPLICATE DATABASE.
- **LOG_FILE_NAME_CONVERT** : Ce paramètre permet d'indiquer le nom des fichiers de journalisation (fichiers redo log) pour la base de données auxiliaire. Il présente le format LOG_FILE_NAME_CONVERT = '*string1*', '*string2*', où *string1* est le modèle pour les noms de fichier dans la base cible et *string2* est celui utilisé pour la base auxiliaire. Par ailleurs, vous pouvez utiliser la clause LOGFILE dans la commande DUPLICATE DATABASE pour indiquer des noms de fichiers de journalisation.

Définir des paramètres pour les noms de fichier (suite)

Au lieu d'utiliser des paramètres d'initialisation pour contrôler les noms des fichiers, vous pouvez utiliser les techniques suivantes pour renommer les fichiers de journalisation :

- Utilisez la clause LOGFILE de la commande DUPLICATE.
- Configurez les paramètres d'initialisation Oracle Managed Files :
DB_CREATE_FILE_DEST, DB_CREATE_ONLINE_DEST_n ou
DB_RECOVERY_FILE_DEST.

Démarrer l'instance en mode NOMOUNT

- Démarrez l'instance auxiliaire en mode NOMOUNT.
- Créez un fichier de paramètres serveur (SPFILE) à partir du fichier texte de paramètres d'initialisation utilisé pour démarrer l'instance.

```
SQL> startup nomount pfile='$HOME/auxinstance/initAUX.ora'
ORACLE instance started.

Total System Global Area  285212672 bytes
Fixed Size                 1218992 bytes
Variable Size              92276304 bytes
Database Buffers          188743680 bytes
Redo Buffers               2973696 bytes
SQL> create spfile
  2  from pfile='$HOME/auxinstance/initAUX.ora';

File created.
```

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Démarrer l'instance en mode NOMOUNT

Une fois que vous avez créé le fichier texte de paramètres d'initialisation, appelez SQL*Plus pour démarrer l'instance auxiliaire en mode NOMOUNT.

Créez ensuite un fichier de paramètres serveur (SPFILE) à partir du fichier texte. Vous pouvez exécuter la commande CREATE SPFILE avant ou après le démarrage de l'instance. Vous devez créer le fichier SPFILE à l'emplacement par défaut pour éviter d'avoir à indiquer l'option PFILE dans la commande DUPLICATE. Cependant, RMAN arrête l'instance auxiliaire et la redémarre dans le cadre du processus de duplication. Vous devez donc indiquer l'option PFILE si vous n'utilisez pas de fichier SPFILE.

Vérifier la disponibilité des sauvegardes et des fichiers de journalisation archivés

- Des sauvegardes de tous les fichiers de données de la base cible doivent être accessibles sur l'hôte de duplication.
- Il peut s'agir d'une combinaison de sauvegardes complètes et de sauvegardes incrémentielles.
- Les fichiers de journalisation archivés nécessaires à la récupération de la base de données dupliquée doivent être accessibles sur l'hôte de duplication.
- Les fichiers de journalisation archivés peuvent être de trois types :
 - Sauvegardes sur un gestionnaire de support
 - Copies d'image
 - Fichiers de journalisation archivés proprement dits

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Vérifier la disponibilité des sauvegardes et des fichiers de journalisation archivés

Les sauvegardes nécessaires à la restauration des fichiers de données doivent être accessibles sur l'hôte de duplication. Vous n'avez pas besoin d'une sauvegarde totale de la base. Pendant le processus de duplication, RMAN peut, en effet, utiliser une combinaison de sauvegardes complètes et de sauvegardes incrémentielles pour les différents fichiers de données.

Les fichiers de journalisation archivés (archived redo logs) requis pour récupérer la base dupliquée jusqu'à un certain point dans le temps doivent également être accessibles. Il peut s'agir de sauvegardes, de copies d'image ou des fichiers de journalisation archivés proprement dits. Les sauvegardes ou les copies peuvent être transférées vers le disque local du noeud de la base dupliquée ou montées sur un réseau via NFS (Network File System), par exemple.

Allouer des canaux auxiliaires

- Les canaux auxiliaires indiquent une connexion entre RMAN et une instance de base de données auxiliaire.
- Si aucun canal automatique n'est configuré, allouez des canaux auxiliaires :
 - Démarrez RMAN avec une connexion à la base de données cible, à l'instance auxiliaire et au catalogue de restauration, le cas échéant.
 - Allouez au moins un canal auxiliaire au sein du bloc RUN.

```

$ rman target sys/oracle_4U@trgt auxiliary
  sys/oracle_4U@auxdb
RMAN> RUN
      {ALLOCATE AUXILIARY CHANNEL aux1 DEVICE TYPE DISK;
      ALLOCATE AUXILIARY CHANNEL aux2 DEVICE TYPE DISK;
      ...
      DUPLICATE TARGET DATABASE to auxdb; . . .
    
```

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Allouer des canaux auxiliaires

Si aucun canal automatique n'est configuré, allouez manuellement au moins un canal auxiliaire avant d'exécuter la commande DUPLICATE. La commande ALLOCATE AUXILIARY CHANNEL doit figurer au sein du même bloc RUN que la commande DUPLICATE.

Le type de canal indiqué dans la commande ALLOCATE AUXILIARY CHANNEL doit correspondre au support sur lequel sont situées les sauvegardes de la base de données cible.

- Si les sauvegardes résident sur disque, vous pouvez allouer plusieurs canaux pour réduire la durée du processus de duplication.
- Pour les sauvegardes sur bande, vous pouvez indiquer un nombre de canaux égal au nombre de périphériques disponibles.

L'instance auxiliaire doit être démarrée avec l'option NOMOUNT. En outre, la base de données cible doit être montée ou ouverte.

Principe de l'opération de duplication RMAN

Lorsque vous exécutez la commande `DUPLICATE`, RMAN effectue les opérations suivantes :

- 1A. Il crée un fichier de paramètres serveur pour l'instance auxiliaire (pour la duplication à partir d'une base active ou de sauvegardes sans connexion à la cible), *ou*
- 1B. Il effectue une restauration à partir de sauvegardes (pour la duplication à partir d'une base de secours ou à partir de sauvegardes sans connexion à la cible).
2. Il monte la sauvegarde du fichier de contrôle.
3. Pour la duplication à partir de sauvegardes : Il sélectionne les sauvegardes à utiliser pour restaurer les fichiers de données sur l'instance auxiliaire.
4. Il restaure les fichiers de données cible vers la base dupliquée.
5. Il effectue une récupération incomplète à l'aide de l'ensemble des sauvegardes incrémentielles et fichiers de journalisation archivés disponibles.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Principe de l'opération de duplication RMAN

Lorsque vous exécutez la commande `DUPLICATE`, RMAN effectue les opérations indiquées dans la diapositive :

- 1A. RMAN crée un fichier de paramètres serveur par défaut pour l'instance auxiliaire si les conditions suivantes sont remplies :
 - La duplication n'implique pas de base de données de secours.
 - Les fichiers de paramètres serveur ne sont pas dupliqués.
 - L'instance auxiliaire n'est pas démarrée avec un fichier de paramètres serveur.
- 1B. RMAN effectue une restauration à partir de sauvegardes (pour la duplication à partir d'une base de secours ou à partir de sauvegardes sans connexion à la cible).
2. RMAN monte la sauvegarde de fichier de contrôle restaurée ou copiée à partir de la base de données active.
3. Pour la duplication à partir de sauvegardes, RMAN sélectionne dans son référentiel les sauvegardes à utiliser pour restaurer les fichiers de données sur l'instance auxiliaire.
4. RMAN restaure les fichiers de données à dupliquer et les copie.
5. RMAN effectue une récupération des fichiers de données à l'aide des sauvegarde incrémentielles et des fichiers de journalisation archivés, jusqu'à un point dans le temps. Cette récupération est nécessaire, même si aucun point dans le temps n'est défini explicitement. En effet, les fichiers de journalisation en ligne de la base source ne sont pas sauvegardés et ne peuvent pas être appliqués à la base de données dupliquée. Le point de récupération extrême de la base dupliquée correspond au fichier de journalisation archivé le plus récent de la base source.

Principe de l'opération de duplication RMAN

Lorsque vous exécutez la commande `DUPLICATE`, RMAN effectue les opérations suivantes :

6. Il arrête et redémarre l'instance auxiliaire en mode `NOMOUNT`.
7. Il crée un nouveau fichier de contrôle qui définit et stocke le nouveau `DBID` dans les fichiers de données.
8. Il ouvre la base de données dupliquée avec l'option `RESETLOGS`.
9. Il crée les fichiers de journalisation en ligne pour la base dupliquée.

Remarque : Après réexécution, la duplication de base de données tente de reprendre à partir du point de défaillance.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Principe de l'opération de duplication RMAN (suite)

6. RMAN arrête et redémarre l'instance auxiliaire en mode `NOMOUNT`.
7. Il crée un nouveau fichier de contrôle qui définit et stocke le nouveau `DBID` unique dans les fichiers de données de la base dupliquée.
8. RMAN ouvre la base de données dupliquée avec l'option `RESETLOGS`.
9. RMAN crée les fichiers de journalisation en ligne pour la base dupliquée.

Remarque : Si la commande `DUPLICATE DATABASE` échoue, vous pouvez réexécuter la commande `DUPLICATE DATABASE`. Le processus de duplication tente alors de reprendre à partir du point de défaillance.

Indiquer des options pour la commande DUPLICATE

Vous pouvez indiquer les options suivantes avec la commande DUPLICATE :

Option	Objectif
SKIP READONLY	Exclure les tablespaces en lecture seule
SKIP TABLESPACE	Exclut les tablespaces nommés
TABLESPACE	Inclut les tablespaces nommés
NOFILENAMECHECK	Empêche la vérification des noms de fichier
OPEN RESTRICTED	Active automatiquement le privilège RESTRICTED SESSION

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Indiquer des options pour la commande DUPLICATE

Indiquez des options supplémentaires lors de l'exécution de la commande DUPLICATE.

SKIP READONLY : Utilisez cette option pour exclure les fichiers de données des tablespaces en lecture seule.

SKIP TABLESPACE : Utilisez cette option pour exclure les tablespaces de la base de données cible. Vous ne pouvez pas exclure le tablespace SYSTEM ni les tablespaces contenant des segments d'annulation (de type undo ou rollback).

TABLESPACE : Utilisez cette option pour inclure les tablespaces de la base de données cible.

NOFILENAMECHECK : Utilisez cette option pour empêcher RMAN de vérifier si des fichiers de données de la base cible portant le même nom que des fichiers de données de la base dupliquée sont utilisés. Vous devez indiquer cette option lorsque des fichiers de journalisation ainsi que des fichiers de la base cible et de la base dupliquée utilisent les mêmes noms. En général, vous l'utilisez lorsqu'une base est dupliquée sur un hôte qui présente la même configuration de disque, la même structure de répertoires et les mêmes noms de fichier que l'hôte de la base cible. Si vous n'indiquez pas l'option NOFILENAMECHECK dans ce cas, RMAN renvoie une erreur.

OPEN RESTRICTED : Utilisez cette option pour activer automatiquement le privilège RESTRICTED SESSION une fois la base de données ouverte.

Utiliser les options supplémentaires de la commande DUPLICATE

Option	Objectif
NOREDO	Signale à RMAN que l'application des fichiers de journalisation doit être supprimée lors de la récupération. Doit être utilisée avec la commande DUPLICATE "sans cible" lorsque la base de données cible est en mode NOARCHIVELOG au moment de la sauvegarde. Peut également être utilisée pour déclarer explicitement qu'aucun fichier de journalisation archivé ne doit être appliqué.
UNDO TABLESPACE	Doit être définie lorsque la base de données cible n'est pas ouverte et qu'il n'existe aucune connexion au catalogue de restauration de sorte que RMAN ne vérifie pas les objets détenus par SYS dans le tablespace.

ORACLE

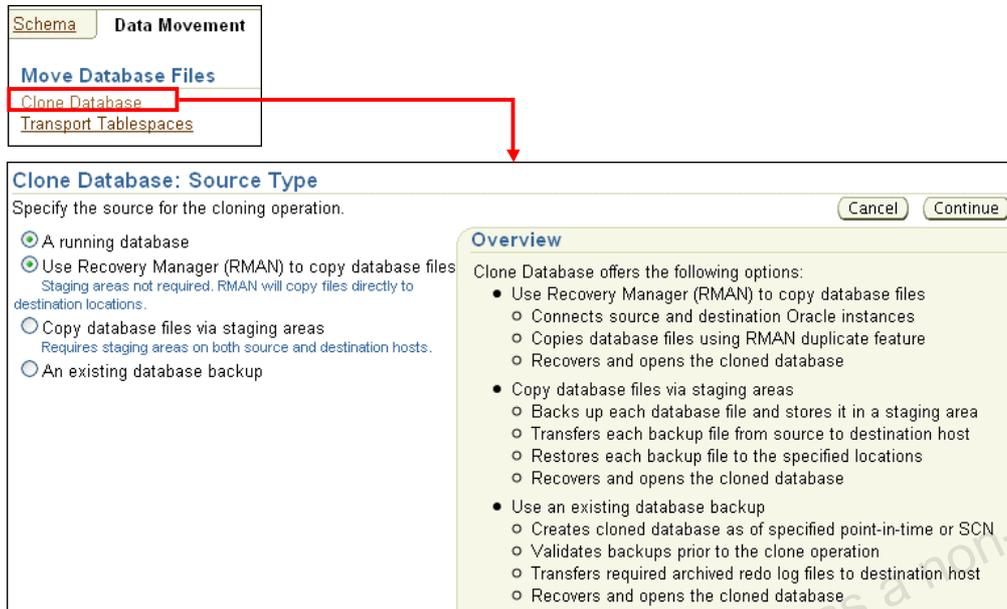
Copyright © 2009, Oracle. Tous droits réservés.

Utiliser les options supplémentaires de la commande DUPLICATE

Les options supplémentaires suivantes ont été introduites dans Oracle Database 11g Release2 pour la commande DUPLICATE :

- **NOREDO** : Cette option est utilisée pour signaler à RMAN que les fichiers de journalisation ne doivent pas être appliqués lors de la phase de récupération de l'opération de duplication. Elle doit être définie lorsque la base de données est en mode NOARCHIVELOG au moment de la sauvegarde ou lorsque les fichiers de journalisation archivés sont indisponibles lors de l'opération de duplication. Cette option est appropriée si une base de données dont le mode actuel est ARCHIVELOG est dupliquée jusqu'à un point dans le temps (lorsqu'elle était en mode NOARCHIVELOG) .
Si vous exécutez une commande DUPLICATE "sans cible" et que la base de données est en mode NOARCHIVELOG, vous devez utiliser cette option pour informer RMAN au sujet du mode de la base. En l'absence de connexion à la base de données cible, RMAN ne peut pas déterminer le mode.
- **UNDO TABLESPACE** : RMAN vérifie qu'aucun objet appartenant à l'utilisateur SYS ne figure dans un des tablespaces dupliqués lors d'une duplication partielle de la base. Les tablespaces SYSTEM, SYSAUX et les tablespaces de segments d'annulation sont exclus de cette vérification. Cependant, si la base de données cible n'est pas ouverte et qu'aucun catalogue de restauration n'est utilisé lors de la duplication, RMAN ne peut pas obtenir les noms des tablespaces d'annulation. Vous devez donc utiliser l'option UNDO TABLESPACE pour fournir ces noms.

Utiliser EM pour cloner une base de données



Copyright © 2009, Oracle. Tous droits réservés.

Utiliser EM pour cloner une base de données

Vous pouvez utiliser Enterprise Manager (EM) pour créer une base de données dupliquée (clone). Dans la page d'accueil EM, sélectionnez Data Movement > Clone database. Vous accédez à la page Clone Database: Source Type.

Pour l'opération de clonage, vous avez le choix entre trois emplacements :

- **Instance en cours d'exécution** : Vous pouvez indiquer une instance en cours d'exécution à cloner.
- **Zone intermédiaire** : Zone de disque définie sur les hôtes source et de destination. Une sauvegarde y est créée et stockée, puis placée dans la zone intermédiaire de destination et lue à partir de l'hôte de destination pour la création de la base clone.
- **Sauvegarde existante** : Si vous disposez d'une sauvegarde reflétant la base de données dans l'état dans lequel vous souhaitez la cloner, vous pouvez l'utiliser.

Quiz

Sélectionnez les affirmations qui sont vraies concernant la duplication de base de données :

1. La duplication d'une base de données peut s'effectuer sans connexion à l'instance auxiliaire.
2. La duplication d'une base de données peut s'effectuer sans connexion au catalogue de restauration.
3. La duplication d'une base de données peut s'effectuer sans connexion à la cible.
4. La duplication d'une base de données nécessite de disposer de sauvegardes RMAN.
5. Vous devez toujours recréer les fichiers de contrôle manuellement sur l'instance auxiliaire.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Réponses : 2, 3

Laurent PERPETUE (laurent.perpetue@orange.fr) has a non-transferable license to use this Student Guide

Synthèse

Ce chapitre vous a permis d'apprendre à :

- énumérer les motifs de création d'une base de données dupliquée
- choisir une technique de duplication
- dupliquer une base de données à l'aide de RMAN
- utiliser une sauvegarde RMAN pour dupliquer une base de données
- dupliquer une base de données à partir d'une instance en cours d'exécution

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.

Présentation de l'exercice 20 : Dupliquer une base de données

Dans cet exercice, vous allez cloner une base de données et recourir à des utilitaires pour configurer une base dupliquée totalement opérationnelle.

ORACLE

Copyright © 2009, Oracle. Tous droits réservés.