

Surveiller et résoudre les conflits de verrouillage

Objectifs

A la fin de ce chapitre, vous pourrez :

- **détecter et résoudre les conflits de verrouillage**
- **gérer les "verrous mortels"**

Verrous externes

- Ils empêchent plusieurs sessions de modifier les mêmes données en même temps.
- Ils sont acquis automatiquement au niveau le plus bas possible pour une instruction donnée.

Transaction 1



Transaction 2



```
SQL> UPDATE hr.employees
2 SET salary=salary+100
3 WHERE employee_id=100;
```

```
SQL> UPDATE hr.employees
2 SET salary=salary*1.1
3 WHERE employee_id=100;
```

Mécanisme de verrouillage

- **Haut niveau de simultanéité des données**
 - Verrous au niveau ligne pour les insertions, les mises à jour et les suppressions
 - Aucun verrou externe nécessaire pour les interrogations
- **Gestion automatique des files d'attente**
- **Verrous détenus jusqu'à la fin de la transaction (validation ou annulation)**

Transaction 1



Transaction 2



```
SQL> UPDATE hr.employees
2 SET salary=salary+100
3 WHERE employee_id=100;
```

```
SQL> UPDATE hr.employees
2 SET salary=salary*1.1
3 WHERE employee_id=101;
```

Simultanéité d'accès aux données

| | | |
|---|----------------------|---|
| Heure : 09:00:00 | Transaction 1 | <code>UPDATE hr.employees SET salary=salary+100 WHERE employee_id=100;</code> |
| | Transaction 2 | <code>UPDATE hr.employees SET salary=salary+100 WHERE employee_id=101;</code> |
| | Transaction 3 | <code>UPDATE hr.employees SET salary=salary+100 WHERE employee_id=102;</code> |
| | ... | ... |
| | Transaction x | <code>UPDATE hr.employees SET salary=salary+100 WHERE employee_id=xxx;</code> |

Verrous LMD

Transaction 1

```
SQL> UPDATE employees
  2  SET salary=salary*1.1
  3  WHERE employee_id= 107;
1 row updated.
```

Transaction 2

```
SQL> UPDATE employees
  2  SET salary=salary*1.1
  3  WHERE employee_id= 106;
1 row updated.
```

Chaque transaction LMD doit acquérir deux verrous :

- **Un verrou de type Row Exclusive pour la ou les lignes mises à jour**
- **Un verrou sur table de type Share, pour la table contenant les lignes**


Mécanisme de mise en file d'attente

Le mécanisme de mise en file d'attente effectue le suivi :

- Des sessions en attente de verrous
- Du mode de verrouillage demandé
- De l'ordre dans lequel les sessions ont demandé le verrou



Conflits de verrouillage

| Transaction 1 | Heure | Transaction 2 |
|--|---|---|
| <pre>UPDATE hr.employees SET salary=salary+100 WHERE employee_id=100; 1 row updated.</pre> | 9:00:00 | <pre>UPDATE hr.employees SET salary=salary+100 WHERE employee_id=101; 1 row updated.</pre> |
| <pre>UPDATE hr.employees SET COMMISSION_PCT=2 WHERE employee_id=101; La session patiente en file d'attente en raison d'un conflit de verrouillage.</pre> | 9:00:05  | <pre>SELECT sum(salary) FROM hr.employees; SUM(SALARY) ----- 692634</pre> |
| <p>La session patiente toujours !</p> | 16:30:00 | <p>Nombreuses opérations de sélection, d'insertion, de mise à jour et de suppression au cours des dernières 7,5 heures, mais pas de validation ni d'annulation !</p> |
| <pre>1 row updated. La session se poursuit.</pre> | 16:30:01 | <pre>commit;</pre> |

Causes possibles des conflits de verrouillage

- **Modifications non validées**
- **Transactions longues**
- **Niveaux de verrouillage inutilement élevés**



Détecter les conflits de verrouillage

Sélectionnez **Blocking Sessions** dans la page Performance.

Database: dba10g > Blocking Sessions

Blocking Sessions

Page Refreshed Nov 25, 2003 6:53:21 PM

[View Session](#) [Kill Session](#)

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

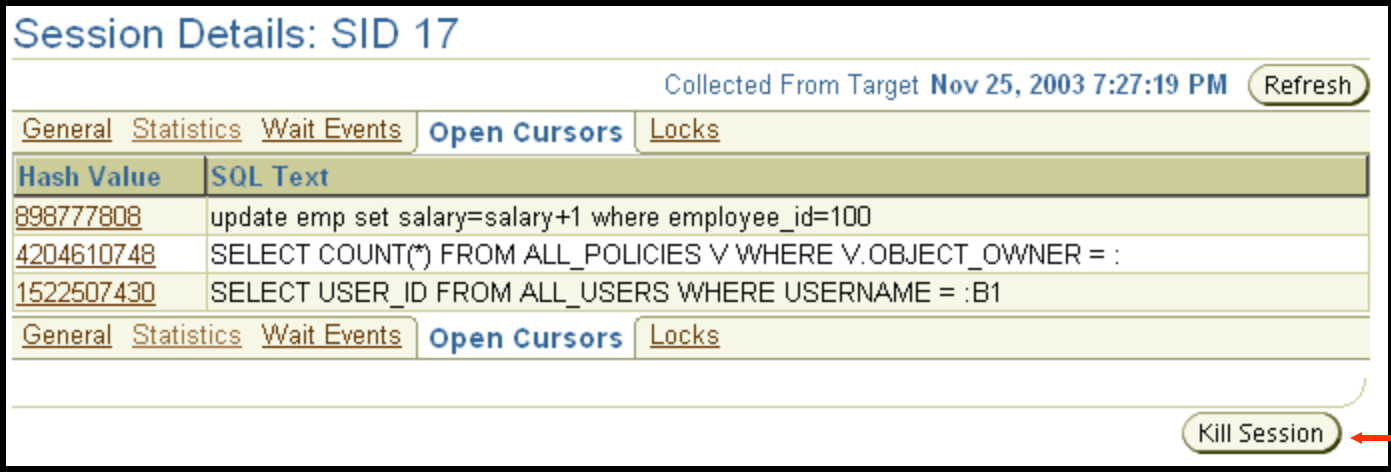
| Select | Username | Sessions Blocked | Session ID | Session Serial Number | SQL Hash Value | Wait Class | Wait Event | P1 | P2 | P3 | Seconds in Wait |
|----------------------------------|-------------------|------------------|--------------------|-----------------------|----------------|-------------|---|------------|---------|----|-----------------|
| <input type="radio"/> | Blocking Sessions | | | | | | | | | | |
| <input checked="" type="radio"/> | HR | 1 | 17 | 28225 | | Idle | SQL*Net message from client | 1413697536 | 1 | 0 | 44 |
| <input type="radio"/> | HR | 0 | 32 | 13948 | 898777808 | Application | enq: TX - row lock contention | 1415053318 | 5373958 | 39 | 35 |

Cliquez sur le lien **Session ID** afin d'afficher des informations sur la session qui a provoqué le verrouillage, y compris l'instruction SQL concernée.

Résoudre les conflits de verrouillage

Pour résoudre un conflit de verrouillage :

- Provoquez la validation ou l'annulation de la transaction par la session détenant le verrou.
- En dernier recours, mettez fin à la session détenant le verrou.



Session Details: SID 17

Collected From Target Nov 25, 2003 7:27:19 PM Refresh


General Statistics Wait Events **Open Cursors** Locks

| Hash Value | SQL Text |
|----------------------------|--|
| 898777808 | update emp set salary=salary+1 where employee_id=100 |
| 4204610748 | SELECT COUNT(*) FROM ALL_POLICIES V WHERE V.OBJECT_OWNER = : |
| 1522507430 | SELECT USER_ID FROM ALL_USERS WHERE USERNAME = :B1 |

General Statistics Wait Events **Open Cursors** Locks

Kill Session

"Verrous mortels"

| Transaction 1 |  | Transaction 2 |
|---|--|--|
| <pre>UPDATE employees SET salary = salary x 1.1 WHERE employee_id = 1000;</pre> | 9:00 | <pre>UPDATE employees SET manager = 1342 WHERE employee_id = 2000;</pre> |
| <pre>UPDATE employees SET salary = salary x 1.1 WHERE employee_id = 2000;</pre> | 9:15 | <pre>UPDATE employees SET manager = 1342 WHERE employee_id = 1000;</pre> |
| ORA-00060: Deadlock detected while waiting for resource | 9:16 | |

Synthèse

Ce chapitre vous a permis d'apprendre à :

- **détecter et résoudre les conflits de verrouillage**
- **gérer les "verrous mortels"**

Exercice 17 : Verrous dans la base de données Oracle

Cet exercice porte sur les tâches administratives courantes liées aux verrous dans Oracle Database 10g, notamment :

- **détecter la session qui provoque le conflit de verrouillage**
- **résoudre les conflits de verrouillage**