

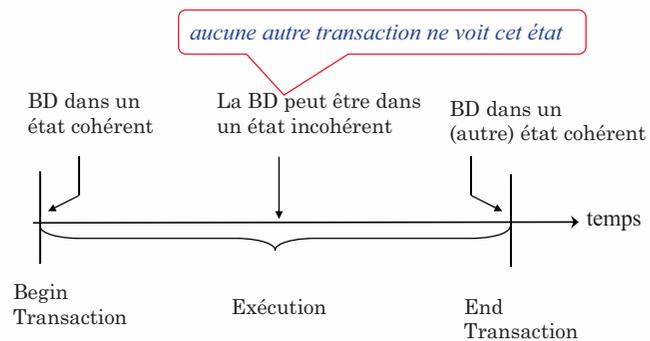
Gestion de transactions

- Définition
- Exemples
- Propriétés des transactions
- Fiabilité et tolérance aux pannes
- Journaux
- Protocoles de journalisation
- Points de reprise

Transaction

Ensemble d'actions qui réalisent des transformations cohérentes de la BD

- opérations de lecture ou d'écriture de données, appelées *granules* (tuples, pages, etc.)



Exemple de transaction

Stock (numP, v, s)

- v : nombre d'exemplaires vendus
- s : nombre d'exemplaires en stock

Commande (numC, numP, ...)

Ne pas vendre plus d'exemplaires v que la quantité en stock s

Transaction Achat(*client*, *produit*)

```

Begin
  select v from Stock where numP=produit
  if (v < s) then
    update Stock set v = v+1 where numP = produit
    insert into Commande values(client, produit, ...)
    Commit;
  else Rollback;
End
    
```

Fiabilité

Problème:

Comment maintenir

atomicité

durabilité

des transactions

5

Types de pannes

Rollback d'une transaction

- Normal : if ou dû à un interblocage
- Anormal : Arrêt de l'appli: environ 3% des cas

Panne système

- panne de processeur, mémoire, alimentation, ...
- le contenu de la mémoire principale est perdu mais disque ok

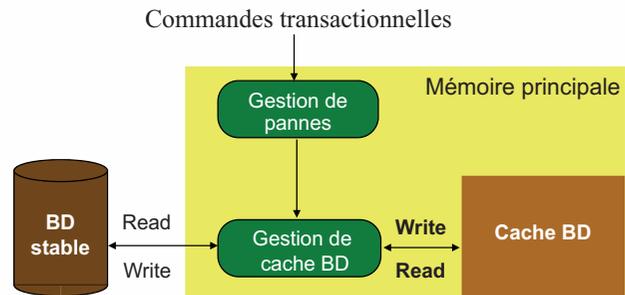
Panne disque

- panne matérielle
- les données de la BD sur disque sont perdues

↓
gravité

6

Architecture pour la gestion de pannes



7

Stratégies de mise-à-jour

Mise-à-jour en place

- chaque mise-à-jour cause la modification de données dans des pages dans le cache BD
- l'ancienne valeur est écrasée par la nouvelle

Mise-à-jour hors-place

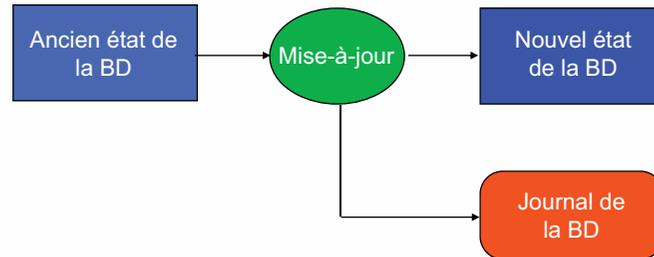
- les nouvelles valeurs de données sont écrites séparément des anciennes dans des pages ombres
- peu utilisé en pratique car très cher
- mises-à-jour des index compliquée

8

Journalisation

Chaque action d'une transaction doit

1. réaliser l'action
2. écrire un enregistrement dans le journal



9

Journal de la BD

Le journal contient les informations nécessaires à la restauration d'un état cohérent de la BD

- Identifiant de transaction
- Type d'opération (action)
- Granules accédés par la transaction pour réaliser l'action
- Ancienne valeur de granule (*image avant*)
- Nouvelle valeur de granule (*image après*)
- ...

Fichier en ajout seulement (append only)

10

Structure du journal

Structure d'un enregistrement :

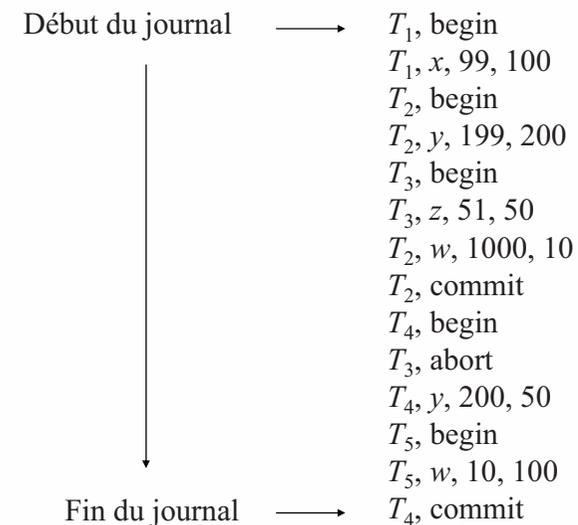
- N° transaction (Trid)
- Type enregistrement {début, update, insert, commit, abort}
- TupleId (rowid sous Oracle)
- [Attribut modifié, Ancienne valeur, Nouvelle valeur] ...

Problème de taille

- on tourne sur N fichiers de taille fixe
- possibilité d'utiliser un fichier haché sur Trid/Tid

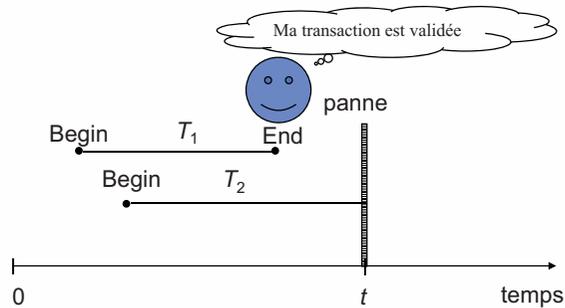
11

Exemple de journal



12

Pourquoi journaliser?

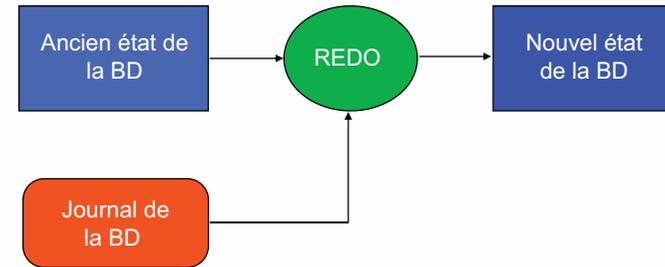


Lors de la reprise

- toutes les mises-à-jour de T_1 doivent être faites dans la BD (REDO)
- aucune mise-à-jour de T_2 ne doit être faite dans la BD (UNDO)

13

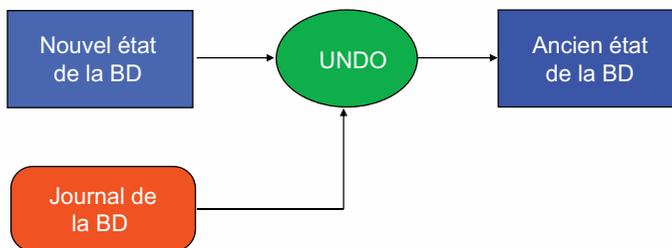
Protocole REDO



L'opération REDO utilise l'information du journal pour refaire les actions qui ont été exécutées ou interrompues
Elle génère la nouvelle image

14

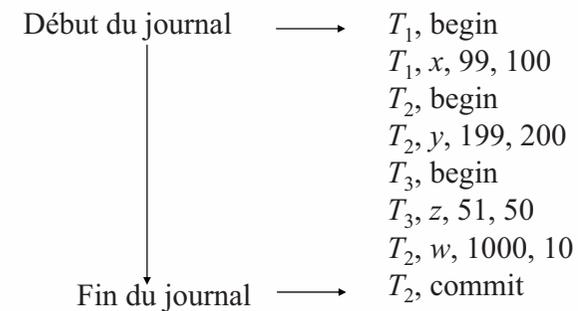
Protocole UNDO



L'opération UNDO utilise l'information du journal pour restaurer l'image avant du granule. Faite en principe avant le REDO

15

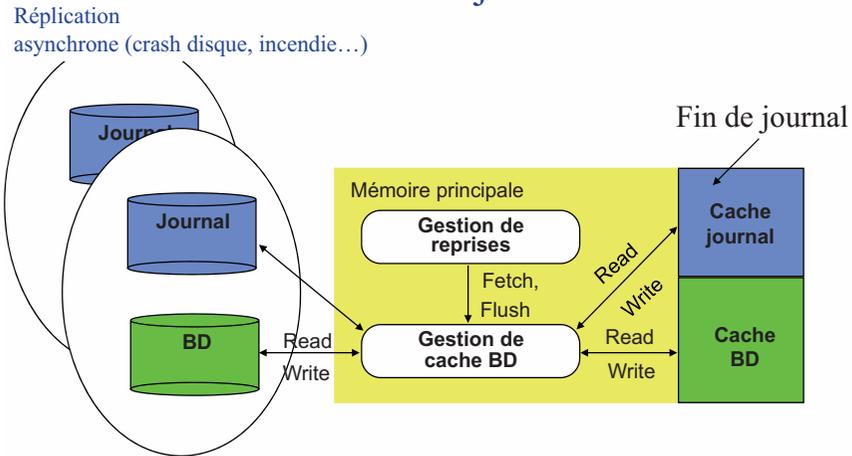
UNDO: parcours vers l'arrière, REDO : parcours vers l'avant



UNDO : T_2 rien (marquée pour Redo), $z:=51$, $x:=99$
REDO : $y:=200$, $w:=10$

16

Interface du journal



17

Gestion du cache BD

Le cache améliore les performances du système, mais a des répercussions sur la reprise (dépend de la politique de migration du cache vers le disque).

Pour simplifier le travail de reconstruction, on peut :

- Empêcher des migrations
 - FIX = Ne rien migrer *pendant* la transaction
- Forcer la migration en fin de transaction
 - FLUSH = Doit migrer à chaque commit

Fix et flush facilite le recouvrement mais contraignent la gestion du cache

18

Gestion du cache BD

Impact sur la reprise :

- **No-fix/no-flush** : UNDO/REDO

Undo nécessaire car les écritures de transactions non validées ont peut être été écrites sur disque et donc rechargées à la reprise.

Redo nécessaire car les écritures de transactions validées n'ont peut être pas été écrites sur disque

- **Fix/no-flush** : REDO
- **No-fix/flush** : UNDO
- **Fix/Flush** : rien à faire

19

Ecriture du **journal** sur disque

Synchrone (forcée): à chaque ajout d'un enregistrement

- ralentit la transaction
- facilite le recouvrement

Asynchrone : périodique ou quand le buffer est plein ou...

- Au plus tard quand la transaction valide

20

Quand écrire le journal sur disque?

Supposons une transaction T qui modifie la page P

Cas chanceux

- le système écrit P dans la BD sur disque
- le système écrit le journal sur disque pour cette opération
- PANNE!... (avant la validation de T)

Nous pouvons reprendre (undo) en restaurant P à son ancien état grâce au journal

Cas malchanceux

- le système écrit P dans la BD sur disque
- PANNE!... (avant l'écriture du journal)

Nous ne pouvons pas récupérer car il n'y a pas d'enregistrement avec l'ancienne valeur dans le journal

Solution: le protocole **Write-Ahead Log (WAL)**

21

Protocole WAL

Observation:

- si la panne précède la validation de transaction, alors toutes ses opérations doivent être défaites, en restaurant les images avant (*partie undo* du journal)
- dès qu'une transaction a été validée, certaines de ses actions doivent pouvoir être refaites, en utilisant les images après (*partie redo* du journal)

Protocole WAL:

- Avant d'écrire dans la BD sur disque, la partie *undo* du journal doit être écrite sur disque
- lors de la validation de transaction, la partie *redo* du journal doit être écrite sur disque avant la mise-à-jour de la BD sur disque

22

Points de reprise

Réduit la quantité de travail à refaire ou défaire lors d'une panne

Un point de reprise enregistre une liste de transactions actives

Pose d'un point de reprise:

- écrire un enregistrement `begin_checkpoint` dans le journal
- écrire les pages du journal et de la BD sur disque
- écrire un enregistrement `end_checkpoint` dans le journal

Remarque :

- Procédure similaire pour rafraichissement des sauvegardes

23

Procédures de reprise

Reprise à chaud

- perte de données en mémoire, mais pas sur disque
- à partir du dernier point de reprise, déterminer les transactions
 - validées : REDO
 - non validées : UNDO

Reprise à froid

- perte de données sur disque
- à partir de la dernière sauvegarde et du dernier point de reprise, faire REDO des transactions validées
- UNDO inutile

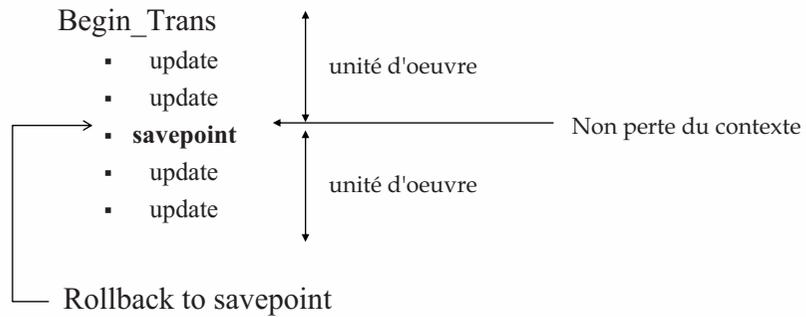
Il peut y avoir des pannes pendant la procédure de reprise....

24

Points de Sauvegardes

Introduction de points de sauvegarde intermédiaires

- (savepoint, commitpoint)



25

Conclusion

- Reprise sur panne = solution pour garantir la durabilité
- Perspectives :
 - Durabilité plus ou moins forte selon les données et la transaction.
 - Exploiter la réplication pour mettre en place la durabilité
 - Utiliser le journal pour interroger un état de la base à une date antérieure quelconque (flash back query).
 - Stockage des données dans une structure inspirée d'un journal
 - Log-Structured Merge tree
 - Utilisé dans la plupart des systèmes de stockage répartis

26