

Bases de Datos

Recuperación de fallos usando bitácora y checkpoints



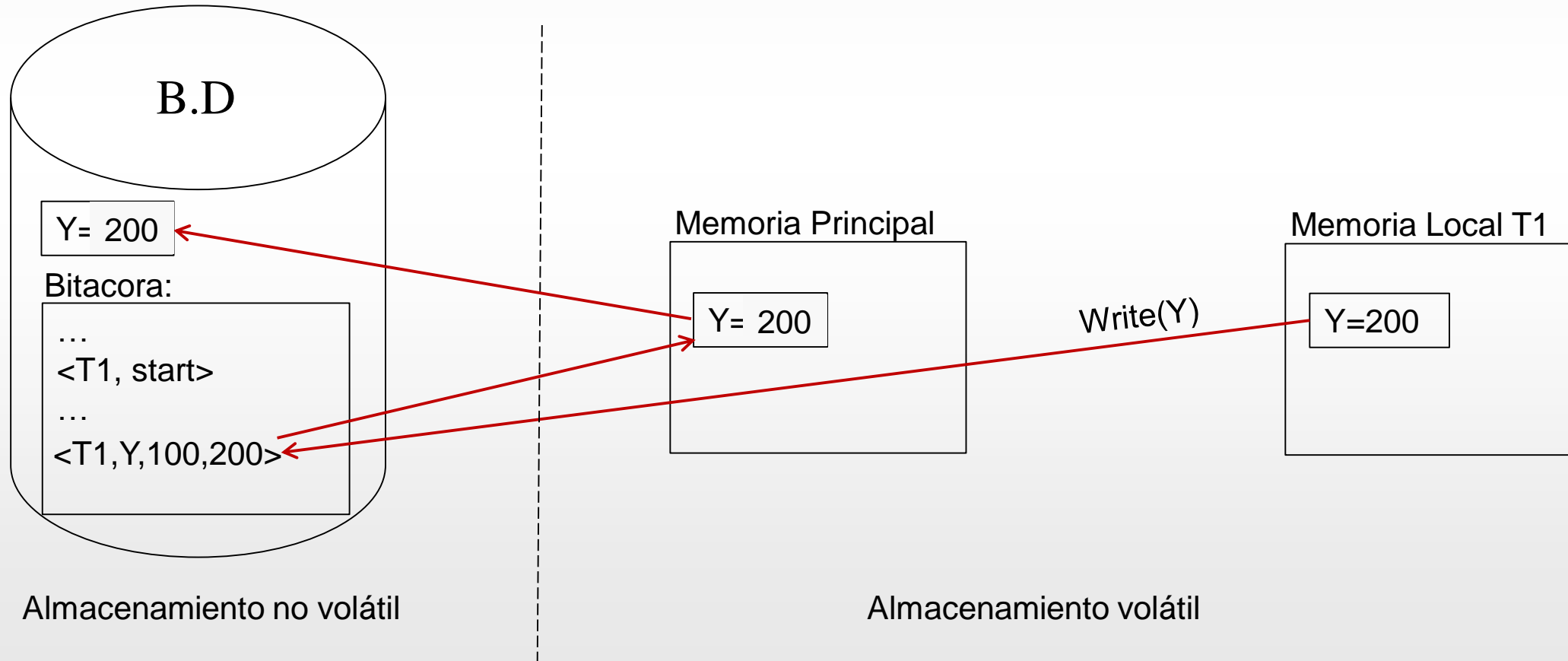
Dr. Diego R. Garcia

**DEPARTAMENTO DE CIENCIAS E
INGENIERÍA DE LA COMPUTACIÓN
UNIVERSIDAD NACIONAL DEL SUR**



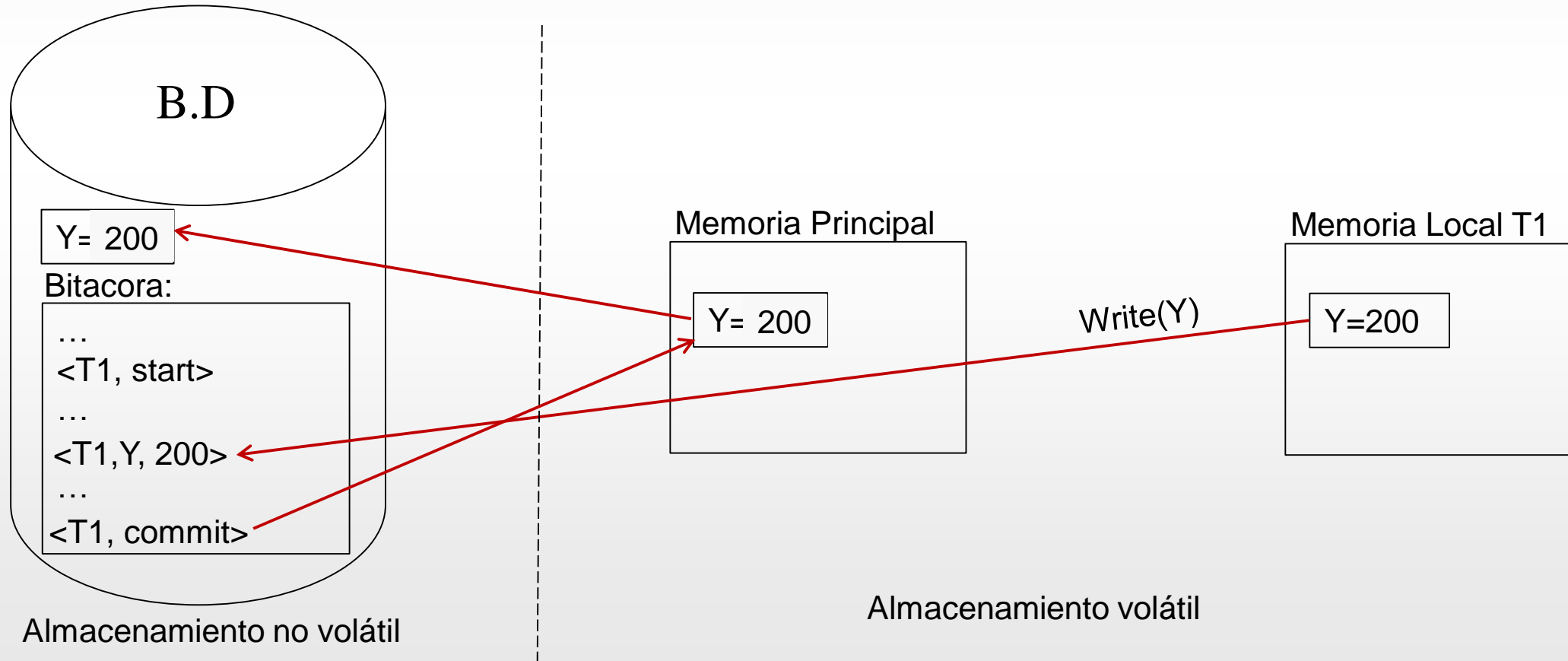
Modificación inmediata

- Las **escrituras sobre la Base de Datos se realizan mientras la transacción está activa.**
- En la bitácora se almacenan los registros $\langle T_i, \text{start} \rangle$ y $\langle T_i, \text{commit} \rangle$ cuando una transacción T_i comienza y termina respectivamente
- Si una transacción T_i escribe Y , primero se almacena en la bitácora el registro: $\langle T_i, Y, \text{valor_viejo}, \text{valor_nuevo} \rangle$ y luego se modifica el valor de Y .



Modificación diferida


- Todas las escrituras de una transacción se registran en la bitácora, pero **solo se modifican los datos en la B.D. después que la transacción comente.**
- Se almacenan en bitácora los registros $\langle T_i, \text{start} \rangle$ y $\langle T_i, \text{commit} \rangle$ cuando T_i comienza y termina.
- Cuando una transacción T_i escribe Y , solo se almacena en la bitácora: $\langle T_i, Y, \text{valor_nuevo} \rangle$.
- Recién cuando la transacción comete se actualizan en la B.D. los valores de los datos que escribió.



Modificación inmediata: ejemplo

Bitácora:

```
...
1. <T0, A, 10, 15>
2. <T1, start>
3. <T0, C, 100, 200>
4. <T1, A, 15, 30>
5. <checkpoint,[T0,T1]>
6. <T2, start>
7. <T2, A , 30, 40>
8. <T3, start>
9. <T0, commit>
10. <T3, B , 11, 21>
11. <T1, B , 21, 31>
12. <T4, start>
13. <T4, B , 31, 41>
14. <checkpoint, [T1,T2,T3,T4]>
15. <T2, C, 200, 300>
16. <T5, start>
17. <T2, commit>
18. <T4, A, 40, 50>
19. <T5, C, 300, 350>
20. <T4, commit>
21. Fallo del sistema
```



- Checkpoint en instante 14:
 1. Grabar en memoria estable los registros de bitácora
 2. Grabar en la B.D. **los datos que están en memoria principal**
 3. Grabar en la bitácora el registro **<checkpoint, [T1,T2,T3,T4]>**
Donde T1, T2, T3 y T4 son las transacciones activas (que no cometieron) al momento de hacer el checkpoint.
- Se recorre hacia atrás la bitácora hasta el último checkpoint (1ero que se encuentra) y se construyen 2 listas:
 - **Undo-List**=**[T5 , T1 , T3]** Transacciones que comenzaron antes (están en la lista de trans. activas) o después del último checkpoint, pero **no** cometieron después del último checkpoint.
 - **Redo-List**= **[T4, T2]** transacciones que cometieron después del último checkpoint.

Modificación inmediata: ejemplo

Bitácora:

```
...
1. <T0, A, 10, 15>
2. <T1, start> ✓
3. <T0, C, 100, 200>
4. <T1, A, 15, 30>
5. <checkpoint,[T0,T1]>
6. <T2, start> ✓
7. <T2, A, 30, 40>
8. <T3, start> ✓
9. <T0, commit>
10. <T3, B, 11, 21>
11. <T1, B, 21, 31>
12. <T4, start> ✓
13. <T4, B, 31, 41>
14. <checkpoint, [T1,T2,T3,T4]>
15. <T2, C, 200, 300>
16. <T5, start> ✓
17. <T2, commit>
18. <T4, A, 40, 50>
19. <T5, C, 300, 350>
20. <T4, commit>
21. Fallo del sistema
```

- Se construyen 2 listas:
 - Undo-List=[T5, T1, T3]
 - Redo-List= [T4, T2]
- Acciones de recuperación:

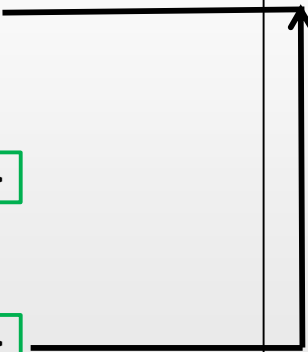
Dato	Valor	Reg. bitácora asociado	Acción asociada
C	300	19.<T5, C, 300,350>	UNDO(T5)
B	21	11.<T1, B, 21,31>	UNDO(T1)
B	11	10.<T3, B, 11,21>	UNDO(T3)
A	15	4.<T1, A, 15, 30>	UNDO(T1)
A	40	7.<T2, A, 30, 40>	REDO(T2)
B	41	13.<T4, B, 31, 41>	REDO(T4)
C	300	15.<T2, C, 200, 300>	REDO(T2)
A	50	18.<T4, A, 40, 50>	REDO(T4)

Valores finales de los datos después de la recuperación:
A=50, B=41 y C=300

Modificación diferida: ejemplo

Bitácora:

```
...
1. <T0, A, 15>
2. <T1, start>
3. <T0, C, 200>
4. <T1, A, 30>
5. <checkpoint>
6. <T2, start>
7. <T2, A, 40>
8. <T3, start>
9. <T0, commit>
10. <T3, B, 21>
11. <T1, B, 31>
12. <T4, start>
13. <T4, B, 41>
14. <checkpoint>
15. <T2, C, 300>
16. <T5, start>
17. <T2, commit>
18. <T4, A, 50>
19. <T5, C, 350>
20. <T4, commit>
21. Fallo del sistema
```



- Checkpoint en instante 14:
 1. Grabar en memoria estable los registros de bitácora.
 2. Grabar en la B.D. **los datos de las transacciones cometidas.**
 3. Grabar en la bitácora el registro **<checkpoint>**
(no es necesario almacenar la lista de transacciones activas: sino cometieron antes del checkpoint, no modificaron la B.D. y no es necesario deshacerlas)
- Se recorre hacia atrás la bitácora y se construyen una sola lista:
 - **Redo-List**= [**T4** , **T2**] con las transacciones que cometieron después del último checkpoint.

Modificación diferida: ejemplo

Bitácora:

```
...
1. <T0, A, 15>
2. <T1, start>
3. <T0, C, 200>
4. <T1, A, 30>
5. <checkpoint>
6. <T2, start> ✓
7. <T2, A, 40>
8. <T3, start>
9. <T0, commit>
10. <T3, B, 21>
11. <T1, B, 31>
12. <T4, start> ✓
13. <T4, B, 41>
14. <checkpoint>
15. <T2, C, 300>
16. <T5, start>
17. <T2, commit>
18. <T4, A, 50>
19. <T5, C, 350>
20. <T4, commit>
21. Fallo del sistema
```

- Se construyen una única lista:
 - Redo-List= [T4 ,T2]
- Acciones de recuperación:

Dato	Valor	Reg. bitácora asociado	Acción asociada
A	40	7.<T2, A, 40>	REDO(T2)
B	41	13.<T4, B, 41>	REDO(T4)
C	300	15.<T2, C, 300>	REDO(T2)
A	50	18.<T4, A, 50>	REDO(T4)

Valores finales de los datos después de la recuperación:
A=50, B=41, C=300