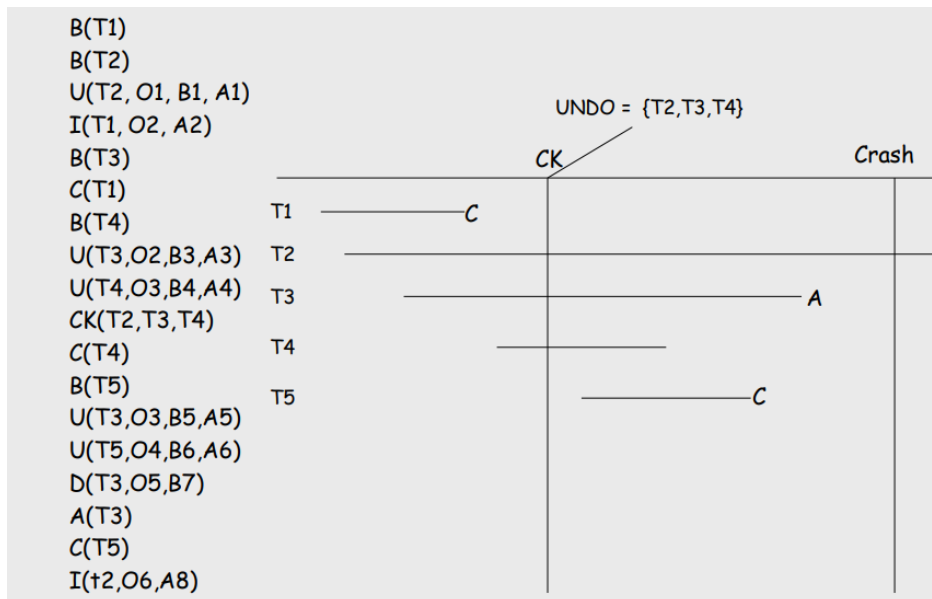


- in commit ma non per forza completate (può servire *REDO*)
- senza commit (vanno annullate, *UNDO*)

La ripresa a caldo quindi:

- trova l'ultimo checkpoint ripercorrendo il log a ritroso
- costruisce gli insiemi *UNDO/REDO*
- ripercorre il log all'indietro fino alla più vecchia fra le transazioni *UNDO/REDO*, disfacendo tutte le azioni delle transazioni in *UNDO*
- ripercorre il log in avanti, rifacendo tutte le azioni delle transazioni in *REDO*

Un esempio pratico:

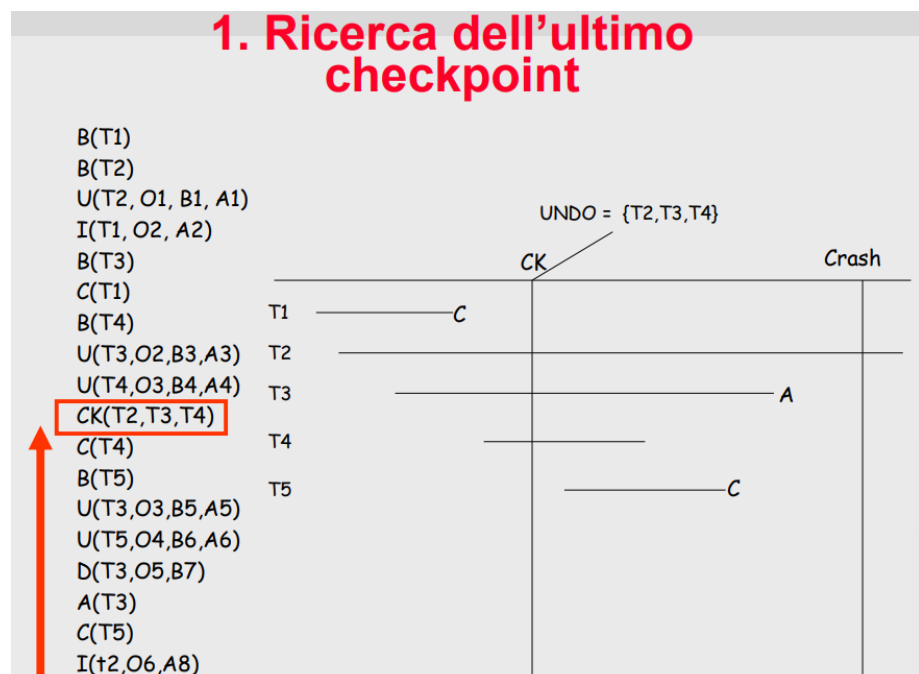


Glossario

iniziali:

I = Insert  
D = Delete  
B = Begin  
C = Commit  
U = Update  
A = Abort  
CK = Check

1) Si cerca l'ultimo checkpoint, partendo da sotto e arrivando a:



2) Si costruiscono gli insiemi UNDO e REDO (nota: questi due insiemi si costruiscono partendo dal basso e arrivando al primo CK/CHECK. Le operazioni vanno comunque rifatte per UNDO e REDO su tutto il file, ma si considerano solo le operazioni che coinvolgono questi due insiemi. Quindi, se devo costruire UNDO/REDO non mi serve scorrere tutto il file di log, ma solo arrivare all'ultimo checkpoint, partendo appunto dal basso). All'interno di UNDO andremo ad inserire:

- inizialmente T2 e T3, dato che su di esse pone controllo il check
- si ha il commit di T4 in C(T4) e la transazione viene posta in REDO
- qualsiasi altra operazione (D/A/I/U) prevede che la transazione rimanga in UNDO; è il caso di T5 che inizia (B/BEGIN), di T3 che aggiorna l'oggetto O3 in B5, di T5 che aggiorna l'oggetto O4 in B6, la cancellazione dell'oggetto O5 in B7 e l'abort di A3 (che fanno in modo T2,T3,T5 rimangano in UNDO
- si ha il commit di T5 e alla fine si rimane con:  
C(T5) → UNDO = {T2,T3}. REDO = {T4, T5}

## 2. Costruzione degli insiemi UNDO e REDO

B(T1)	0. UNDO = {T2,T3,T4}. REDO = {}
B(T2)	
8. U(T2, O1, B1, A1)	1. C(T4) → UNDO = {T2, T3}. REDO = {T4}
I(T1, O2, A2)	
B(T3)	2. B(T5) → UNDO = {T2,T3,T5}. REDO = {T4} Setup
C(T1)	
B(T4)	3. C(T5) → UNDO = {T2,T3}. REDO = {T4, T5}
7. U(T3,O2,B3,A3)	
9. U(T4,O3,B4,A4)	
CK(T2,T3,T4)	
1. C(T4)	
2. B(T5)	
6. U(T3,O3,B5,A5)	
10. U(T5,O4,B6,A6)	
5. D(T3,O5,B7)	
A(T3)	
3. C(T5)	
4. I(T2,O6,A8)	

3) Dato l'insieme degli UNDO, si ripercorre dalla fine all'inizio tutto il log, rifacendo le operazioni che riguardano T2 e T3. Quindi:

- si ha un INSERT di T2 su O6; questo dovrà essere cancellato (quindi quando si ha un INSERT si ha la cancellazione, per permettere di rifarla → D(O6)
- si ha il DELETE di O5 in B7 e dunque → O5 = B7 (che era lo stato precedente, ma è la convenzione che si adotta)
- si ha un UPDATE DI T3 su O3 e → O3=B5
- prima del CHECK si ha un UPDATE di T3 su O2 e → O2=B3
- finalmente, si ha un UPDATE di T2 su O1 in B1 → O1=B1

## 3. Fase UNDO

B(T1)	0. UNDO = {T2,T3,T4}. REDO = {}
B(T2)	
8. U(T2, O1, B1, A1)	1. C(T4) → UNDO = {T2, T3}. REDO = {T4}
I(T1, O2, A2)	
B(T3)	2. B(T5) → UNDO = {T2,T3,T5}. REDO = {T4} Setup
C(T1)	
B(T4)	3. C(T5) → UNDO = {T2,T3}. REDO = {T4, T5}
7. U(T3,O2,B3,A3)	4. D(O6)
9. U(T4,O3,B4,A4)	5. O5 = B7
CK(T2,T3,T4)	
1. C(T4)	6. O3 = B5
2. B(T5)	7. O2 = B3
6. U(T3,O3,B5,A5)	8. O1=B1
10. U(T5,O4,B6,A6)	
5. D(T3,O5,B7)	
A(T3)	
3. C(T5)	
4. I(T2,O6,A8)	

4) Per la fase di REDO, si fa la stessa cosa al contrario, dunque partendo dall'inizio e arrivando alla fine.

Sull'insieme si ha da considerare T4 e T5

Si nota che le operazioni che li riguardano sono:

- U(T4,O3,B4,A4) e quindi porta a dire O3 = B4
- U(T5,O4,B6,A6) e quindi porta a dire O4 = A6

## 4. Fase REDO

B(T1)	0. UNDO = {T2,T3,T4}. REDO = {}	
B(T2)		
8. U(T2,O1,B1,A1)	1. C(T4) → UNDO = {T2,T3}. REDO = {T4}	
I(T1,O2,A2)	2. B(T5) → UNDO = {T2,T3,T5}. REDO = {T4}	Setup
B(T3)	3. C(T5) → UNDO = {T2,T3}. REDO = {T4,T5}	
C(T1)		
B(T4)		
7. U(T3,O2,B3,A3)	4. D(O6)	
9. U(T4,O3,B4,A4)	5. O5 = B7	
CK(T2,T3,T4)		
1. C(T4)	6. O3 = B5	Undo
2. B(T5)	7. O2 = B3	
6. U(T3,O3,B5,A5)	8. O1 = B1	
10. U(T5,O4,B6,A6)		
5. D(T3,O5,B7)		
A(T3)	9. O3 = A4	
3. C(T5)		Redo
4. I(T2,O6,A8)	10. O4 = A6	

Sempre seguendo l'ordine, tiene traccia di quali operazioni rifare.

Se il DB è scritto in Modalità Immediata (immediatamente dopo il log), il REDO non è necessario.

Invece se il DB è scritto in Modalità Differita (solo dopo un commit), l'UNDO non è necessario.

È possibile venga fatto l'UNDO oppure il REDO di operazioni con effetto nullo nel database, come se fosse fatto una sola volta (ad esempio:  $\text{undo}(\text{undo}(A)) = \text{undo}(A)$   $\text{redo}(\text{redo}(A)) = \text{redo}(A)$ )

In merito ai *guasti* abbiamo:

- *guasti di sistema (soft)*, intendendo errori di programma, crash di sistema, caduta di tensione; si opera la ripresa a caldo (*warm restart*), ripetendo le transazioni
- *guasti di dispositivo (hard)*, intendendo guasti su dispositivi di memoria secondaria; tuttavia si ha ridondanza su varie memorie stabili del log, non perdendo la memoria stabile

La struttura dei guasti, oltre al checkpoint, periodicamente esegue dei *dump* (copie del database, ad esempio su nastro) di tutto il contenuto del DB. Grazie a questo evita di dover ogni volta ricominciare.