

Gestione dei dati

Parte 4 Esercitazione su gestione del recovery

Maurizio Lenzerini, Riccardo Rosati

Facoltà di Ingegneria
Sapienza Università di Roma
Anno Accademico 2012/2013

<http://www.dis.uniroma1.it/~rosati/gd/>



SAPIENZA
UNIVERSITÀ DI ROMA

Esercizio 1: cold restart

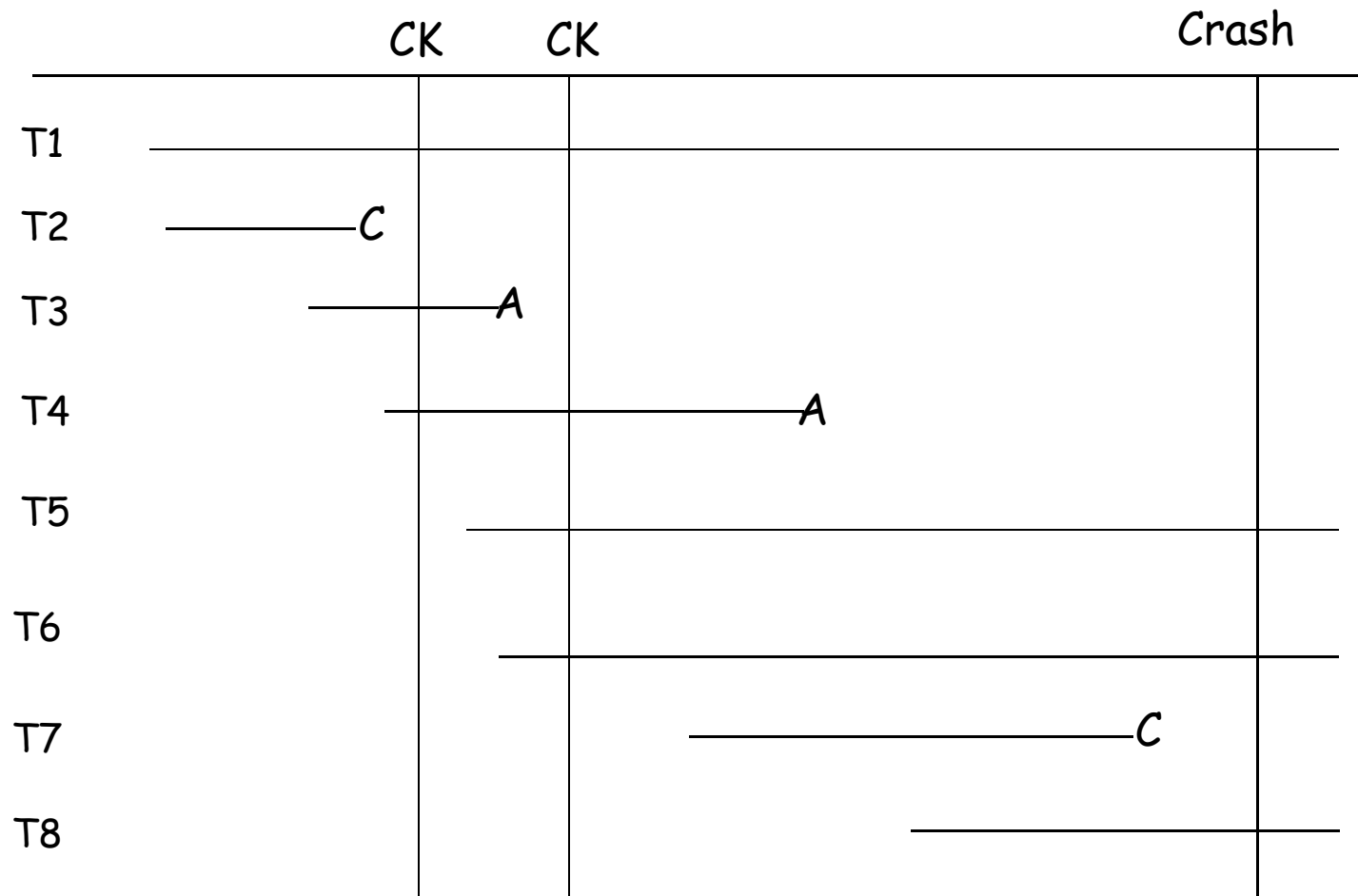
- Si consideri il seguente log: DUMP, B(T1), B(T2), B(T3), I(T1,O1,A1), D(T2,O2,B2), B(T4), U(T4,O3,B3,A3), U(T1,O4,B4,A4), C(T2), CK(T1,T3,T4), B(T5), B(T6), U(T5,O5,B5,A5), A(T3), CK(T1,T4,T5,T6), B(T7), A(T4), U(T7,O6,B6,A6), U(T6,O3,B7,A7), B(T8), C(T7)
- A questo punto della esecuzione il sistema subisce un guasto di dispositivo. Si assuma che la modalità di scrittura nella memoria di massa sia quella mista, e quindi richieda sia undo sia redo.

Soluzione: ricostruzione dal DUMP

- DUMP, B(T1), B(T2), B(T3), I(T1,O1,A1), D(T2,O2,B2), B(T4), U(T4,O3,B3,A3), U(T1,O4,B4,A4), C(T2), CK(T1,T3,T4), B(T5), B(T6), U(T5,O5,B5,A5), A(T3), CK(T1,T4,T5,T6), B(T7), A(T4), U(T7,O6,B6,A6), U(T6,O3,B7,A7), B(T8), C(T7)
- Si accede all'ultimo DUMP (la prima operazione riportata nel log), e si ricopia dal back-up in modo selettivo la parte di basi di dati danneggiata
- Si ripercorre in avanti il log a partire da B(T1), e si eseguono tutte le operazioni fino a C(T7)
- Si effettua un warm restart

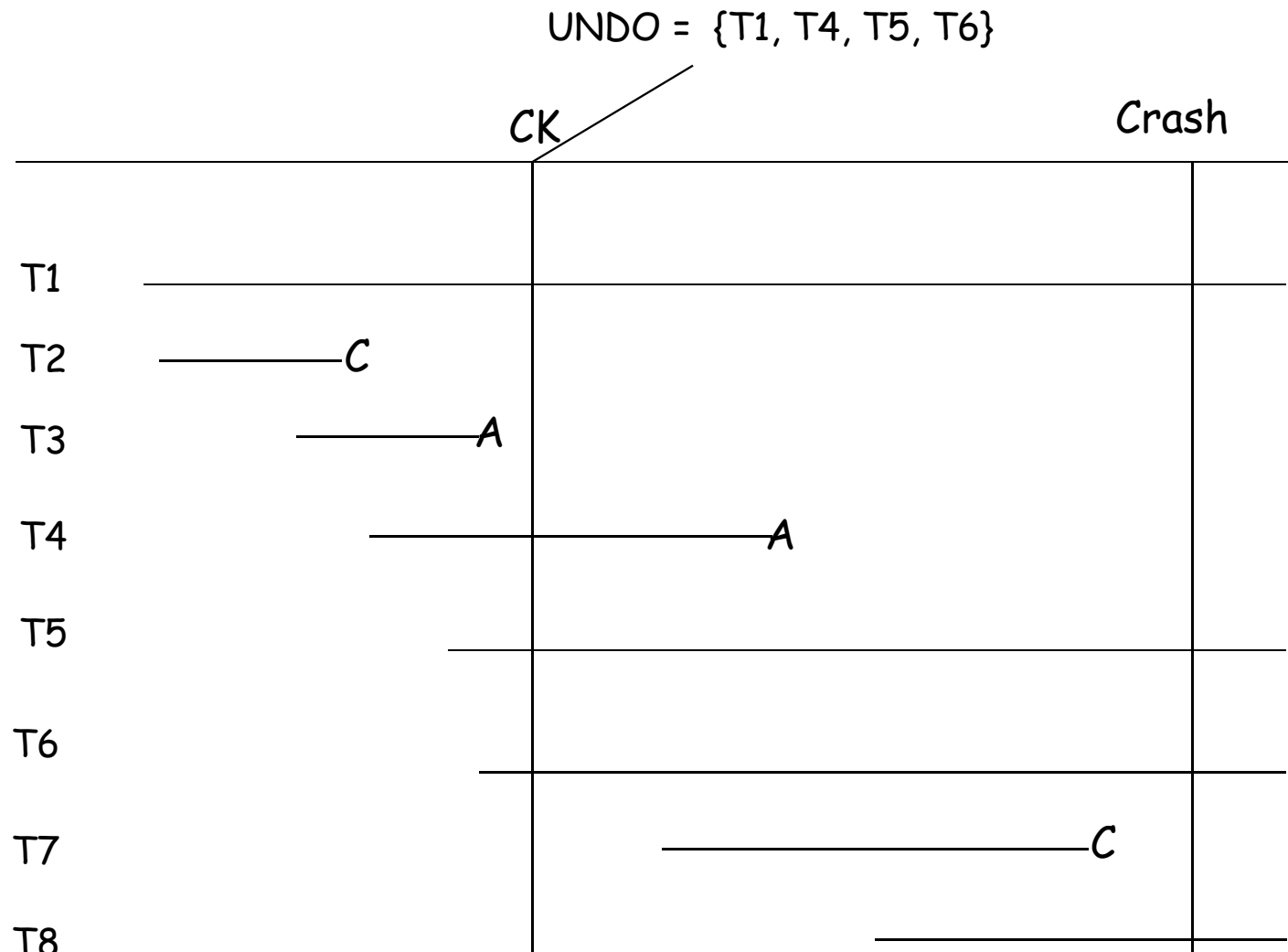
Soluzione: sequenza di warm restart

B(T1),
 B(T2),
 B(T3),
 I(T1,O1,A1),
 D(T2,O2,B2),
 B(T4),
 U(T4,O3,B3,A3),
 U(T1,O4,B4,A4),
 C(T2),
 CK(T1,T3, T4),
 B(T5),
 B(T6),
 U(T5,O5,B5,A5),
 A(T3),
 CK(T1,T4,T5,T6),
 B(T7),
 A(T4),
 U(T7,O6,B6,A6),
 U(T6,O3,B7,A7),
 B(T8),
 C(T7)



Soluzione: checkpoint più recente

B(T1),
B(T2),
B(T3),
I(T1,O1,A1),
D(T2,O2,B2),
B(T4),
U(T4,O3,B3,A3),
U(T1,O4,B4,A4),
C(T2),
CK(T1,T3, T4),
B(T5),
B(T6),
U(T5,O5,B5,A5),
A(T3),
CK(T1,T4,T5,T6),
B(T7),
A(T4),
U(T7,O6,B6,A6),
U(T6,O3,B7,A7),
B(T8),
C(T7)



Soluzione: insiemi UNDO e REDO

B(T1),
B(T2),
B(T3),
I(T1,O1,A1),
D(T2,O2,B2),
B(T4),
U(T4,O3,B3,A3),
U(T1,O4,B4,A4),
C(T2),
CK(T1,T3, T4),
B(T5),
B(T6),
U(T5,O5,B5,A5),
A(T3),
CK(T1,T4,T5,T6),
B(T7),
A(T4),
U(T7,O6,B6,A6),
U(T6,O3,B7,A7),
B(T8),
C(T7)

0. UNDO = {T1, T4, T5, T6}. REDO = {}

1. B(T7) → {T1, T4, T5, T6, T7}. REDO = {}

2. B(T8) → {T1, T4, T5, T6, T7, T8}. REDO = {}

3. C(T7) → {T1, T4, T5, T6, T8}. REDO = {T7}

Soluzione: fase di UNDO

B(T1),
B(T2),
B(T3),
I(T1,O1,A1),
D(T2,O2,B2),
B(T4),
U(T4,O3,B3,A3),
U(T1,O4,B4,A4),
C(T2),
CK(T1,T3, T4),
B(T5),
B(T6),
U(T5,O5,B5,A5),
A(T3),
CK(T1,T4,T5,T6),
B(T7),
A(T4),
U(T7,O6,B6,A6),
U(T6,O3,B7,A7),
B(T8),
C(T7)

0. UNDO = {T1, T4, T5, T6}. REDO = {}

1. B(T7) → {T1, T4, T5, T6, T7}. REDO = {}

2. B(T8) → {T1, T4, T5, T6, T7, T8}. REDO = {}

3. C(T7) → {T1, T4, T5, T6, T8}. REDO = {T7}

4. O3 = B7

5. O5 = B5

6. O4 = B4

7. O3 = B3

8. D(O1)

Undo

Soluzione: fase di REDO

B(T1),
B(T2),
B(T3),
I(T1,O1,A1),
D(T2,O2,B2),
B(T4),
U(T4,O3,B3,A3),
U(T1,O4,B4,A4),
C(T2),
CK(T1,T3, T4),
B(T5),
B(T6),
U(T5,O5,B5,A5),
A(T3),
CK(T1,T4,T5,T6),
B(T7),
A(T4),
U(T7,O6,B6,A6),
U(T6,O3,B7,A7),
B(T8),
C(T7)

0. UNDO = {T1, T4, T5, T6}. REDO = {}

1. B(T7) → {T1, T4, T5, T6, T7}. REDO = {}

2. B(T8) → {T1, T4, T5, T6, T7, T8}. REDO = {}

3. C(T5) → {T1, T4, T5, T6, T8}. REDO = {T7}

4. O3 = B7

5. O5 = B5

6. O4 = B4

7. O3 = B3

8. D(O1)

9. O6 = A6

Undo

Redo

Esercizio 2

Si consideri il seguente file di log:

DUMP, B(T1), B(T2), I(T1,O1,A1), U(T2,O2,B2,A2), B(T3), B(T4),
U(T4,O3,B3,A3), U(T1,O4,B4,A4), C(T2), CK(T1,T3, T4), B(T5),
B(T6), U(T5,O5,B5,A5), U(T3,O2,A2,A2'), A(T3), CK(T1,T4,T5,T6),
B(T7), A(T4), U(T7,O6,B6,A6), U(T6,O3,A3,A3'), B(T8), C(T7)

analizzare il file di log al fine di stabilire se tale file può essere generato da uno scheduler che segue il protocollo 2PL stretto.

si assuma che una azione di update corrisponda ad una coppia di azioni read-write (ad esempio, U(T1,O4,B4,A4) va considerata come la coppia di azioni consecutive r1(O4), w1(O4)) e che le azioni insert e delete corrispondano ad azioni write (ad esempio, I(T1,O1,A1) va considerata come una azione w1(O1))