

# Data Bases II

## Warm Restart

Michele Beretta

[michele.beretta@unibg.it](mailto:michele.beretta@unibg.it)



## Exercise F.1

Describe the warm restart, indicating the progressive construction of the sets UNDO and REDO, and the recovery actions, given the following log configuration:

DUMP,  $b(t_1), b(t_2), b(t_3), i(t_1, o_1, a_1), d(t_2, o_2, b_2),$   
 $b(t_4), u(t_4, o_3, b_3, a_3), u(t_1, o_4, b_4, a_4), c(t_2),$   
CKPT( $t_1, t_3, t_4$ ),  $b(t_5), b(t_6), u(t_5, o_5, b_5, a_5), a(t_3),$   
CKPT( $t_1, t_4, t_5, t_6$ ),  $b(t_7), a(t_4), u(t_7, o_6, b_6, a_6),$   
 $u(t_6, o_3, b_7, a_7), b(t_8), a(t_7),$  FAILURE

Note: we are using the following notation

- $b(t) = t$  begins
- $i(t, o, a) = t$  inserts an object  $o$  with value  $a$
- $u(t, o, b, a) = t$  updates  $o$  from value  $b$  to value  $a$
- $d(t, o, b) = t$  deletes  $o$ , which had value  $b$
- $a(t) = t$  aborts
- $c(t) = t$  commits
- $\text{CKPT}(t_i, \dots, t_j) = t_i$  to  $t_j$  are alive at a checkpoint

How to do it:

1. Trace back the log until the last checkpoint and initialize UNDO and REDO sets:
  - UNDO is composed of all the alive transactions at the CKPT
  - REDO is empty
2. Trace the log forward and
  - Add to UNDO all transactions with a begin record
  - Move from UNDO to REDO all transactions that commit
3. Trace the log backwards to the beginning and undo actions in the UNDO set
4. Reapply actions in the REDO set (same order as the log)

## 1. Find the first checkpoint and initialize sets

DUMP,  $b(t_1), b(t_2), b(t_3), i(t_1, o_1, a_1), d(t_2, o_2, b_2),$   
 $b(t_4), u(t_4, o_3, b_3, a_3), u(t_1, o_4, b_4, a_4), c(t_2),$   
CKPT( $t_1, t_3, t_4$ ),  $b(t_5), b(t_6), u(t_5, o_5, b_5, a_5), a(t_3),$   
**CKPT( $t_1, t_4, t_5, t_6$ )**,  $b(t_7), a(t_4), u(t_7, o_6, b_6, a_6),$   
 $u(t_6, o_3, b_7, a_7), b(t_8), a(t_7), FAILURE$

$$UNDO = \{t_1, t_4, t_5, t_6\}$$

$$REDO = \{\}$$

## 2. Update UNDO and REDO sets

DUMP,  $b(t_1)$ ,  $b(t_2)$ ,  $b(t_3)$ ,  $i(t_1, o_1, a_1)$ ,  $d(t_2, o_2, b_2)$ ,  
 $b(t_4)$ ,  $u(t_4, o_3, b_3, a_3)$ ,  $u(t_1, o_4, b_4, a_4)$ ,  $c(t_2)$ ,  
CKPT( $t_1, t_3, t_4$ ),  $b(t_5)$ ,  $b(t_6)$ ,  $u(t_5, o_5, b_5, a_5)$ ,  $a(t_3)$ ,  
CKPT( $t_1, t_4, t_5, t_6$ ),  $b(t_7)$ ,  $a(t_4)$ ,  $u(t_7, o_6, b_6, a_6)$ ,  
 $u(t_6, o_3, b_7, a_7)$ ,  $b(t_8)$ ,  $a(t_7)$ , FAILURE

UNDO =  $\{t_1, t_4, t_5, t_6, t_7, t_8\}$

REDO =  $\{\}$

### 3. Trace back and undo actions

DUMP,  $b(t_1), b(t_2), b(t_3), i(t_1, o_1, a_1),$   
 $d(t_2, o_2, b_2), b(t_4), u(t_4, o_3, b_3, a_3),$   
 $u(t_1, o_4, b_4, a_4), c(t_2),$   
CKPT( $t_1, t_3, t_4$ ),  $b(t_5),$   
 $b(t_6), u(t_5, o_5, b_5, a_5), a(t_3),$   
CKPT( $t_1, t_4, t_5, t_6$ ),  $b(t_7), a(t_4),$   
 $u(t_7, o_6, b_6, a_6), u(t_6, o_3, b_7, a_7),$   
 $b(t_8), a(t_7), FAILURE$

UNDO =  $\{t_1, t_4, t_5, t_6, t_7, t_8\}$

#### Actions

$o_3 = b_7$

$o_6 = b_6$

$o_5 = b_5$

$o_4 = b_4$

$o_3 = b_3$

delete( $o_1$ )

4. Trace forward and redo actions.

In this case, the REDO set is empty, so we don't have to do anything.

## Exercise F.2

Same as before.

DUMP,  $b(t_1)$ ,  $u(t_1, o_1, b_1, a_1)$ ,  $b(t_2)$ ,  $b(t_3)$ ,  
 $u(t_3, o_3, b_3, a_3)$ ,  $i(t_2, o_2, a_2)$ ,  $c(t_2)$ , CKPT( $t_1, t_3$ ),  
 $c(t_3)$ ,  $b(t_4)$ ,  $u(t_4, o_2, b_4, a_4)$ ,  $u(t_4, o_3, b_5, a_5)$ ,  $b(t_5)$ ,  
 $i(t_5, o_6, a_6)$ ,  $a(t_1)$ ,  $c(t_4)$ ,  $u(t_5, o_7, b_7, a_7)$ ,  
 $d(t_5, o_1, b_8)$ , FAILURE

## 1. Find the first checkpoint and initialize sets

DUMP,  $b(t_1)$ ,  $u(t_1, o_1, b_1, a_1)$ ,  $b(t_2)$ ,  $b(t_3)$ ,  
 $u(t_3, o_3, a_3, b_3)$ ,  $i(t_2, o_2, a_2)$ ,  $c(t_2)$ , **CKPT**( $t_1, t_3$ ),  
 $c(t_3)$ ,  $b(t_4)$ ,  $u(t_4, o_2, b_4, a_4)$ ,  $u(t_4, o_3, b_5, a_5)$ ,  $b(t_5)$ ,  
 $i(t_5, o_6, a_6)$ ,  $a(t_1)$ ,  $c(t_4)$ ,  $u(t_5, o_7, b_7, a_7)$ ,  
 $d(t_5, o_1, b_8)$ , FAILURE

$$\text{UNDO} = \{t_1, t_3\}$$

$$\text{REDO} = \{\}$$

## 2. Update UNDO and REDO sets

DUMP,  $b(t_1)$ ,  $u(t_1, o_1, b_1, a_1)$ ,  $b(t_2)$ ,  $b(t_3)$ ,  
 $u(t_3, o_3, a_3, b_3)$ ,  $i(t_2, o_2, a_2)$ ,  $c(t_2)$ , CKPT( $t_1, t_3$ ),  
 $c(t_3)$ ,  $b(t_4)$ ,  $u(t_4, o_2, b_4, a_4)$ ,  $u(t_4, o_3, b_5, a_5)$ ,  $b(t_5)$ ,  
 $i(t_5, o_6, a_6)$ ,  $a(t_1)$ ,  $c(t_4)$ ,  $u(t_5, o_7, b_7, a_7)$ ,  
 $d(t_5, o_1, b_8)$ , FAILURE

$$\text{UNDO} = \{t_1, t_5\}$$

$$\text{REDO} = \{t_3, t_4\}$$

### 3. Trace back and undo actions

DUMP,  $b(t_1)$ ,  $u(t_1, o_1, b_1, a_1)$ ,  $b(t_2)$ ,  $b(t_3)$ ,  
 $u(t_3, o_3, a_3, b_3)$ ,  $i(t_2, o_2, a_2)$ ,  $c(t_2)$ ,  
CKPT( $t_1, t_3$ ),  $c(t_3)$ ,  $b(t_4)$ ,  $u(t_4, o_2, b_4, a_4)$ ,  
 $u(t_4, o_3, b_5, a_5)$ ,  $b(t_5)$ ,  
 $i(t_5, o_6, a_6)$ ,  $a(t_1)$ ,  $c(t_4)$ ,  
 $u(t_5, o_7, b_7, a_7)$ ,  $d(t_5, o_1, b_8)$ , FAILURE

UNDO =  $\{t_1, t_5\}$

#### Actions

$o_1 = b_8$

$o_7 = b_7$

delete( $o_6$ )

$o_1 = b_1$

#### 4. Trace forward and redo actions

DUMP,  $b(t_1)$ ,  $u(t_1, o_1, b_1, a_1)$ ,  $b(t_2)$ ,  $b(t_3)$ ,

$u(t_3, o_3, b_3, a_3)$ ,  $i(t_2, o_2, a_2)$ ,  $c(t_2)$ ,

CKPT( $t_1, t_3$ ),  $c(t_3)$ ,  $b(t_4)$ ,  $u(t_4, o_2, b_4, a_4)$ ,

$u(t_4, o_3, b_5, a_5)$ ,  $b(t_5)$ ,

$i(t_5, o_6, a_6)$ ,  $a(t_1)$ ,  $c(t_4)$ ,

$u(t_5, o_7, b_7, a_7)$ ,  $d(t_5, o_1, b_8)$ , FAILURE

REDO =  $\{t_3, t_4\}$

Actions

$o_3 = a_3$

$o_2 = a_4$

$o_3 = a_5$