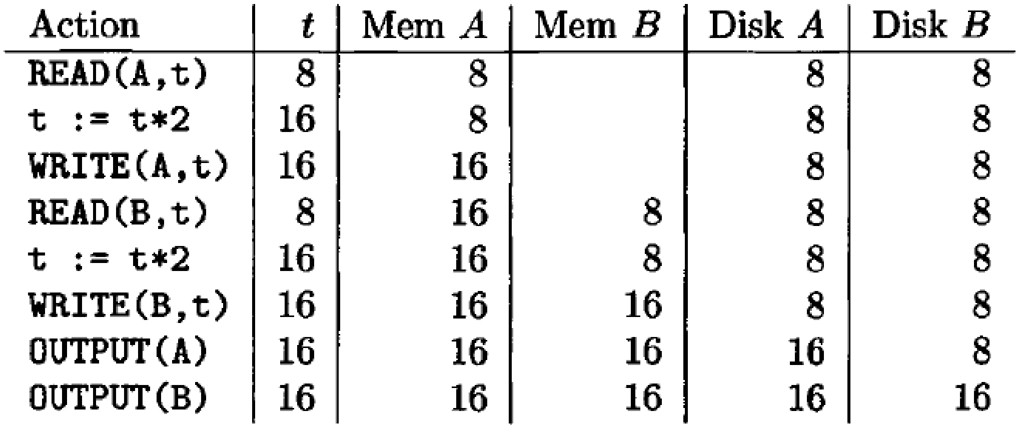
Chapter 17: Coping With System Failures

Failure Modes:

1. Erroneous Data Entry: if entered data is false.
2. Media Failures: if error happen in a drive. Solution: multiple drives
3. Catastrophic Failure: agar zalzaly sai cs department ur jai 😮. Solution: Backup of drive in different places.
4. System Failure: occur due to transaction failure.

ISSUES AND MODELS FOR RESILIENT OPERATION

- Input(x): copy DB element x into buffer.
- Read(x,t): copy DB element x to local variable t.
- Write(x,t): copy value of local variable t to DB element x in buffer.
- Output(x): write DB element x from buffer to hard disk.
- Example is at pdf page 887.



Action	t	Mem A	Mem B	Disk A	Disk B
READ(A,t)	8	8		8	8
t := t*2	16	8		8	8
WRITE(A,t)	16	16		8	8
READ(B,t)	8	16	8	8	8
t := t*2	16	16	8	8	8
WRITE(B,t)	16	16	16	8	8
OUTPUT(A)	16	16	16	16	8
OUTPUT(B)	16	16	16	16	16

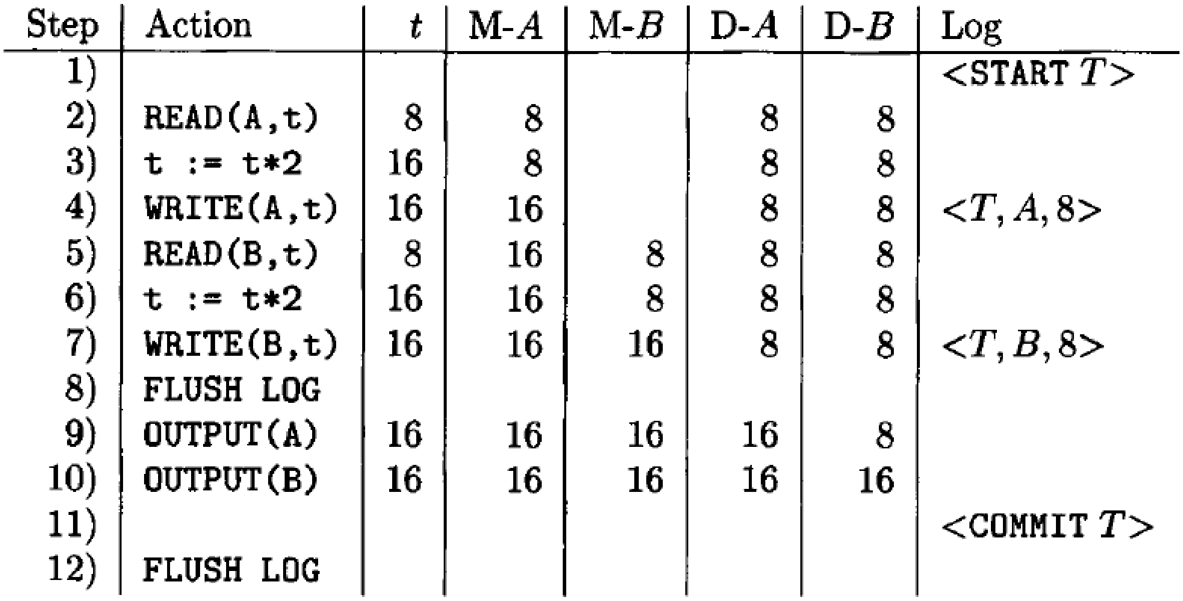
Figure 17.2: Steps of a transaction and its effect on memory and disk

-

Undo Logging:

Rules:

1. Log record of form <T,X,V> must be written to disk before updated values to disk.
2. Updated values must be written to disk before the commit log record to disk.
3. Example is at pdf page 891



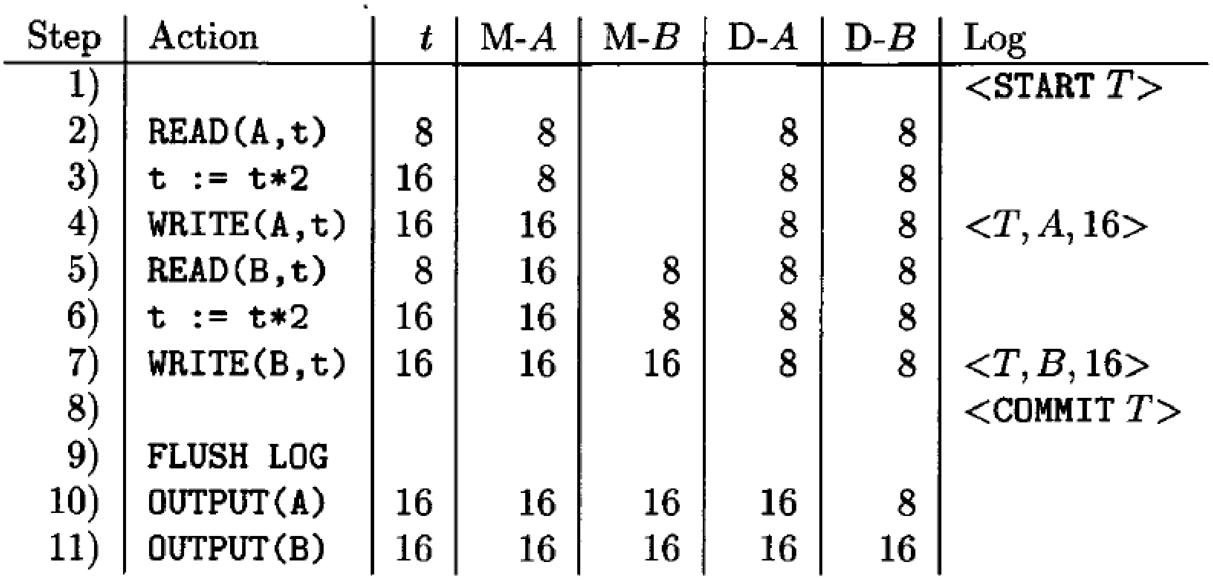
Step	Action	t	M-A	M-B	D-A	D-B	Log
1)							<START T>
2)	READ(A,t)	8	8		8	8	
3)	t := t*2	16	8		8	8	
4)	WRITE(A,t)	16	16		8	8	$<T, A, 8>$
5)	READ(B,t)	8	16	8	8	8	
6)	t := t*2	16	16	8	8	8	
7)	WRITE(B,t)	16	16	16	8	8	$<T, B, 8>$
8)	FLUSH LOG						
9)	OUTPUT(A)	16	16	16	16	8	
10)	OUTPUT(B)	16	16	16	16	16	
11)							<COMMIT T>
12)	FLUSH LOG						

Figure 17.3: Actions and their log entries

4.

Redo Logging:

Rules:

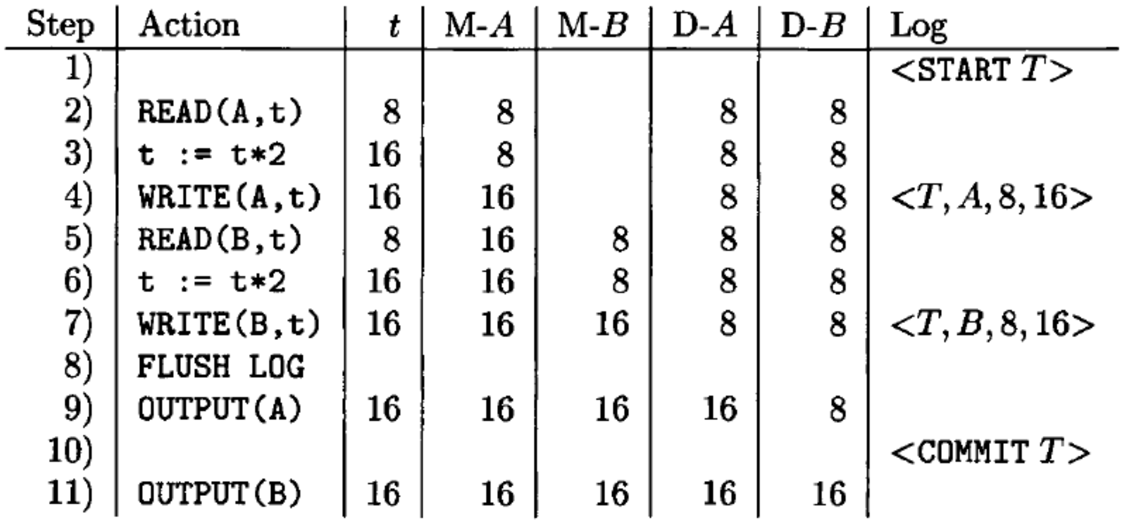
1. All log records(update record, commit record) must be written to disk before updated values to disk.
2. Example is at pdf page 901
3. 

Step	Action	t	M-A	M-B	D-A	D-B	Log
1)							<START T>
2)	READ(A,t)	8	8		8	8	
3)	t := t*2	16	8		8	8	
4)	WRITE(A,t)	16	16		8	8	<T, A, 16>
5)	READ(B,t)	8	16	8	8	8	
6)	t := t*2	16	16	8	8	8	
7)	WRITE(B,t)	16	16	16	8	8	<T, B, 16>
8)							<COMMIT T>
9)	FLUSH LOG						
10)	OUTPUT(A)	16	16	16	16	8	
11)	OUTPUT(B)	16	16	16	16	16	

Figure 17.7: Actions and their log entries using redo logging

Undo/Redo Logging:
Rules:

1. Before modifying DB element x, it is necessary that update record (T,X,V,W) must appear on disk.
2. Example is at pdf page 907



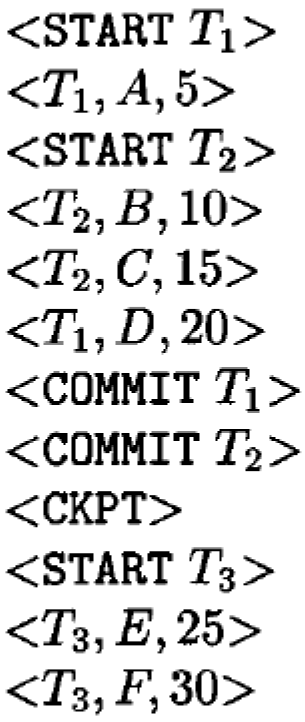
Step	Action	t	M-A	M-B	D-A	D-B	Log
1)							<START T>
2)	READ(A,t)	8	8		8	8	
3)	t := t*2	16	8		8	8	
4)	WRITE(A,t)	16	16		8	8	<$T, A, 8, 16$>
5)	READ(B,t)	8	16	8	8	8	
6)	t := t*2	16	16	8	8	8	
7)	WRITE(B,t)	16	16	16	8	8	<$T, B, 8, 16$>
8)	FLUSH LOG						
9)	OUTPUT(A)	16	16	16	16	8	
10)							<COMMIT T>
11)	OUTPUT(B)	16	16	16	16	16	

3. Figure 17.9: A possible sequence of actions and their log entries using undo/redo

Checkpointing:

Simple Checkpointing: does not allows new transactions to enter the system during the checkpoint.

- Example is at pdf page 896



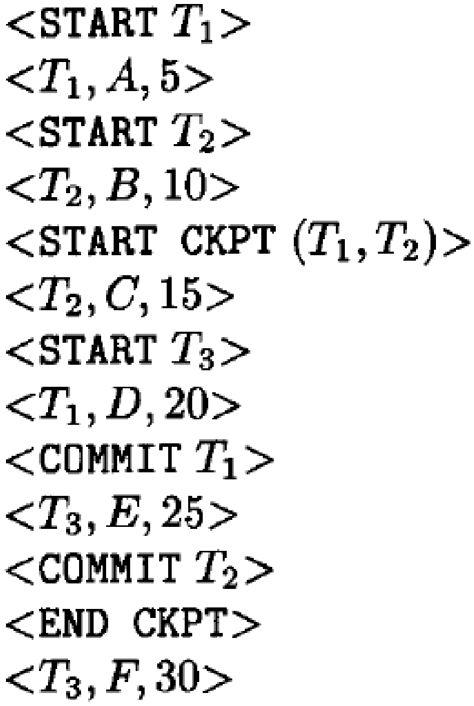
<START T_1>
<$T_1, A, 5$>
<START T_2>
<$T_2, B, 10$>
<$T_2, C, 15$>
<$T_1, D, 20$>
<COMMIT T_1>
<COMMIT T_2>
<CKPT>
<START T_3>
<$T_3, E, 25$>
<$T_3, F, 30$>

Figure 17.4: An undo log

-

Nonquiescent Checkpointing: allows new transactions to enter the system during the checkpoint

- Example is at pdf page 898



<START T_1>
<$T_1, A, 5$>
<START T_2>
<$T_2, B, 10$>
<START CKPT (T_1, T_2)>
<$T_2, C, 15$>
<START T_3>
<$T_1, D, 20$>
<COMMIT T_1>
<$T_3, E, 25$>
<COMMIT T_2>
<END CKPT>
<$T_3, F, 30$>

- Figure 17.5: An undo log using nonquiescent checkpointing

Noted By: Ali Ahmed