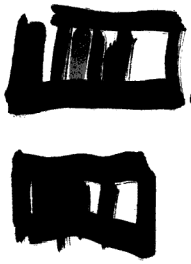
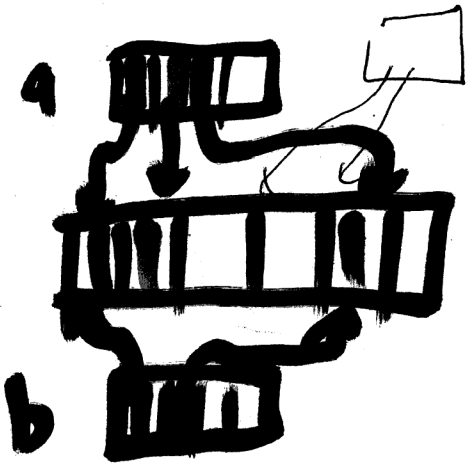


Twin Pages

Pro

spatial Locality

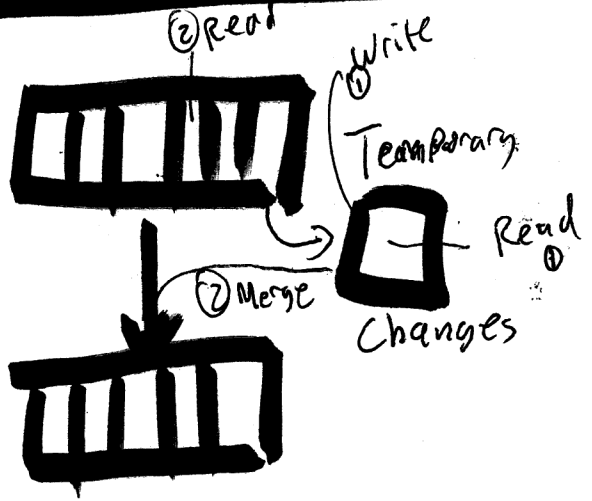


Shadow Paging

Pro

Faster Updates

Less Storage



Before T1

↓

Transaction Running

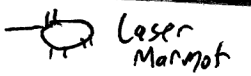
↓

Commit

↓

Merge

↓



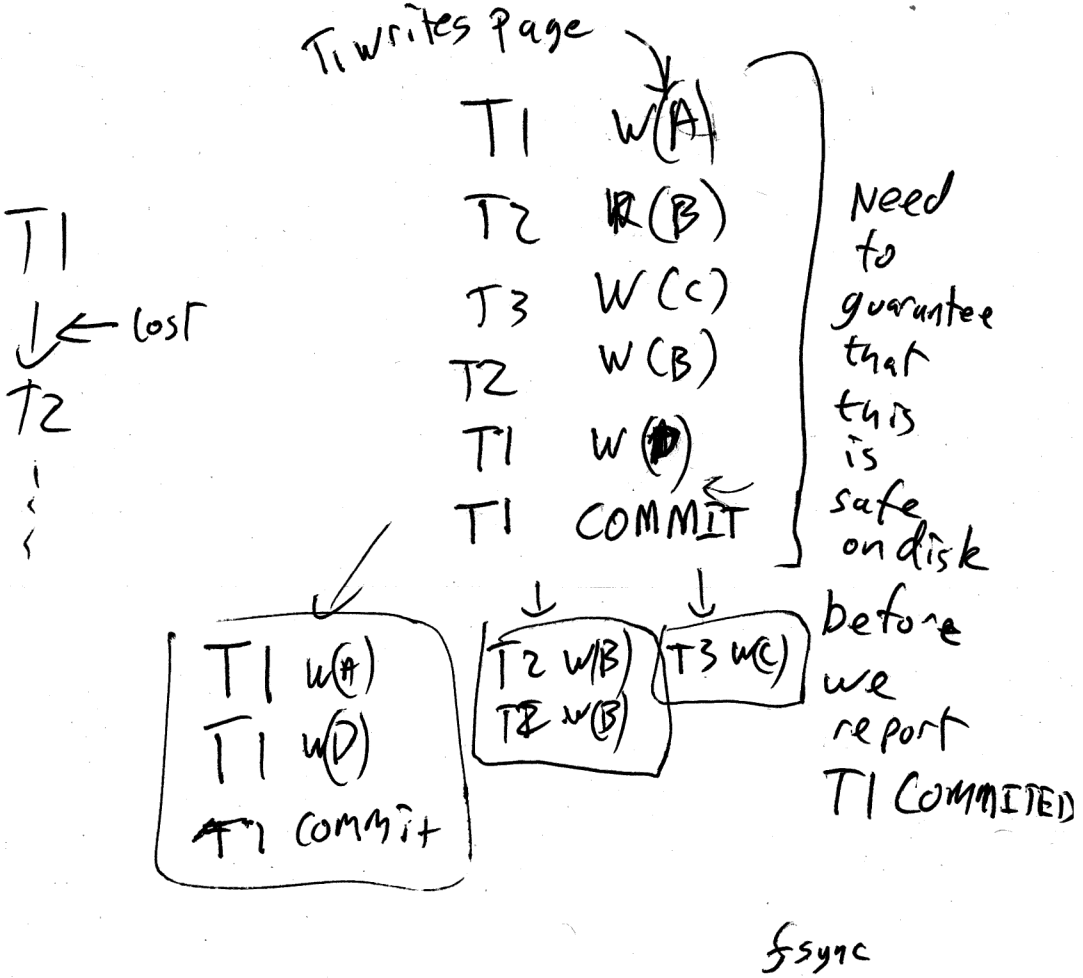
OK

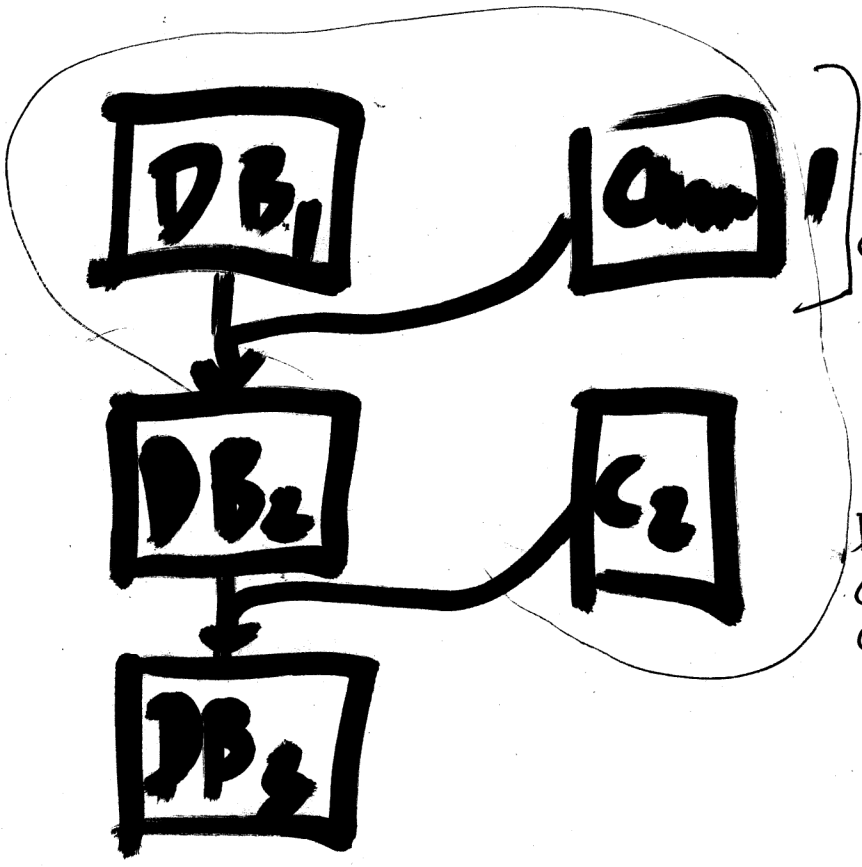
Can abort easily by deleting changes

Naive Twin Paging (Batched)

↳ Lose track of intermediate version

↳ Transactions delayed to end of batch





DB_1 + $Change_1$ = DB_2

DB_1 + $Changes_1$ = DB_2 + $Changes_2$

DB_1 exists ← DB_1

↓

Changes exist ← DB_1 + $changes_1$

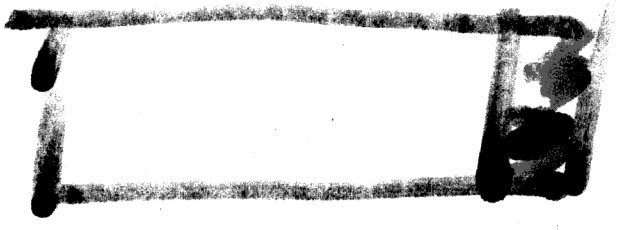
↓

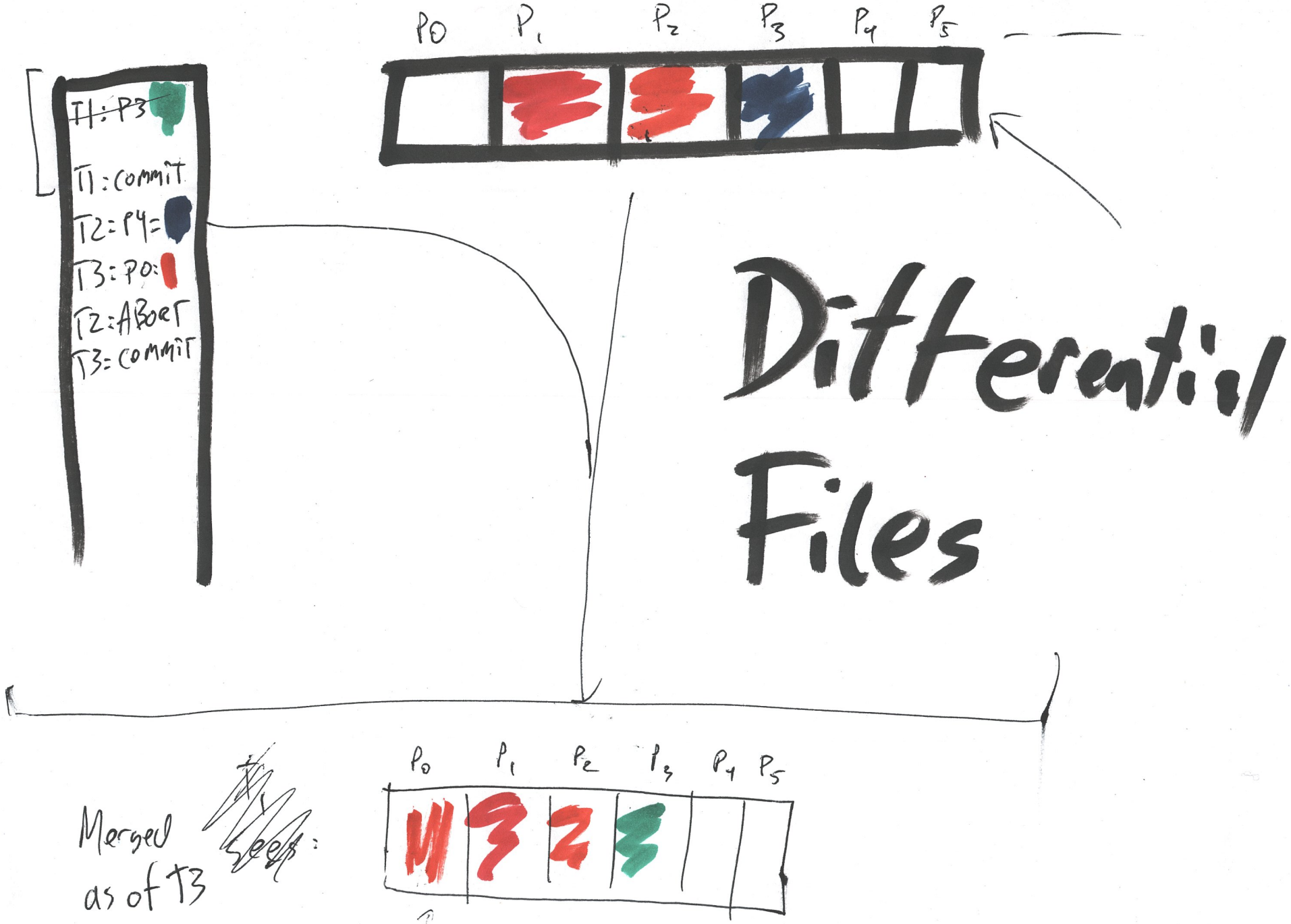
DB2 creation start ← DB_2 ← DB_1 + $changes_1$

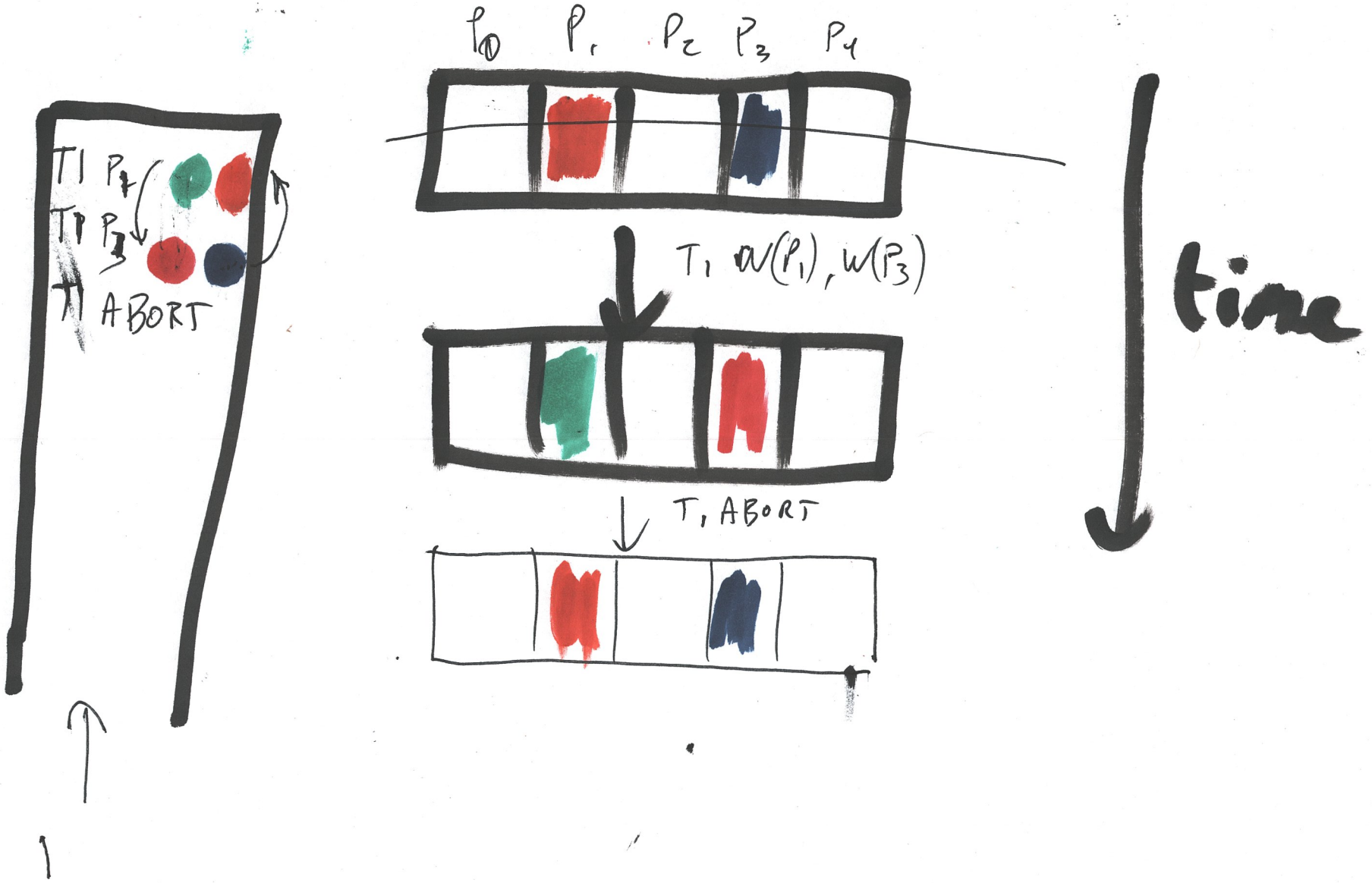
↓

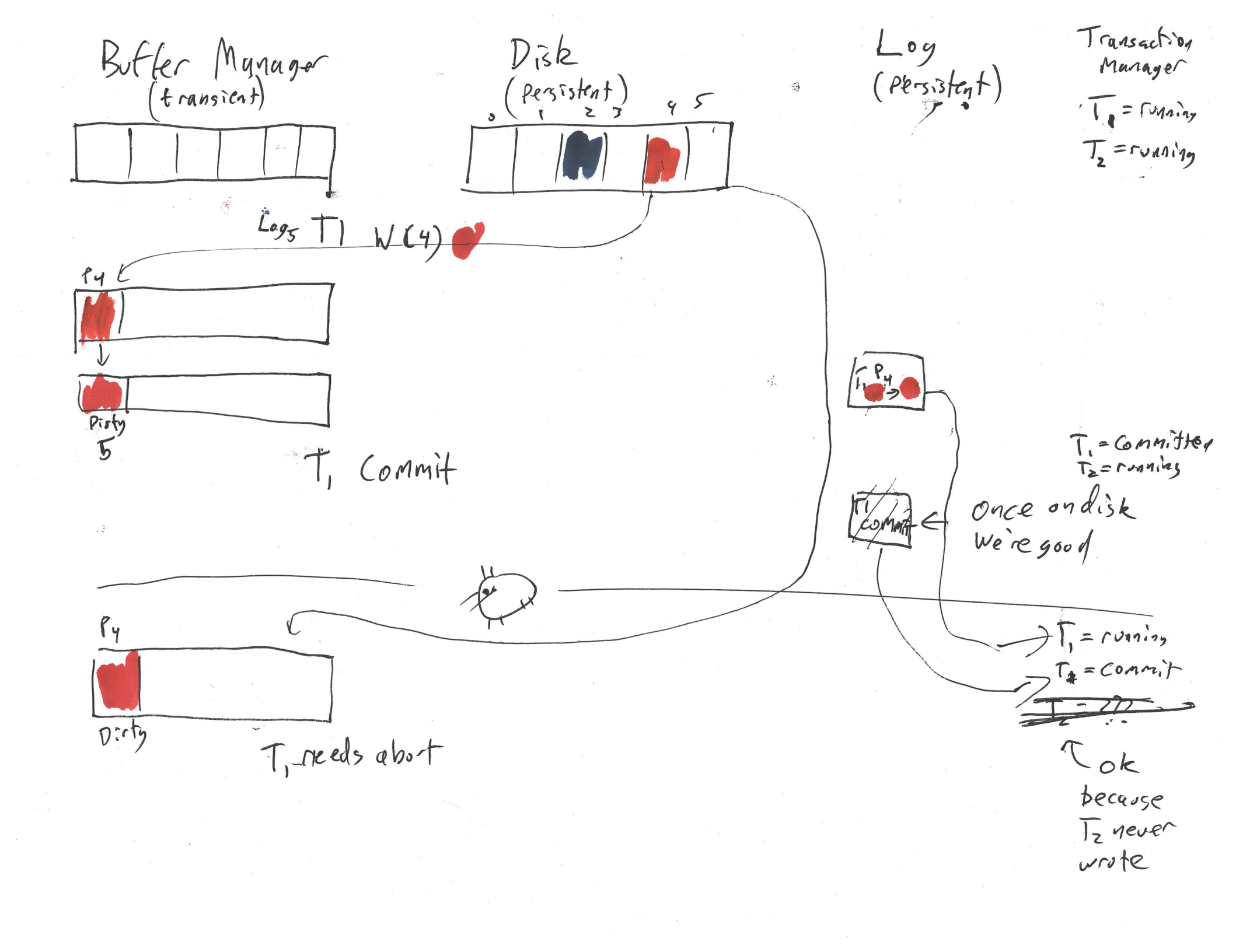
DB2 finishes + $Changes_2$

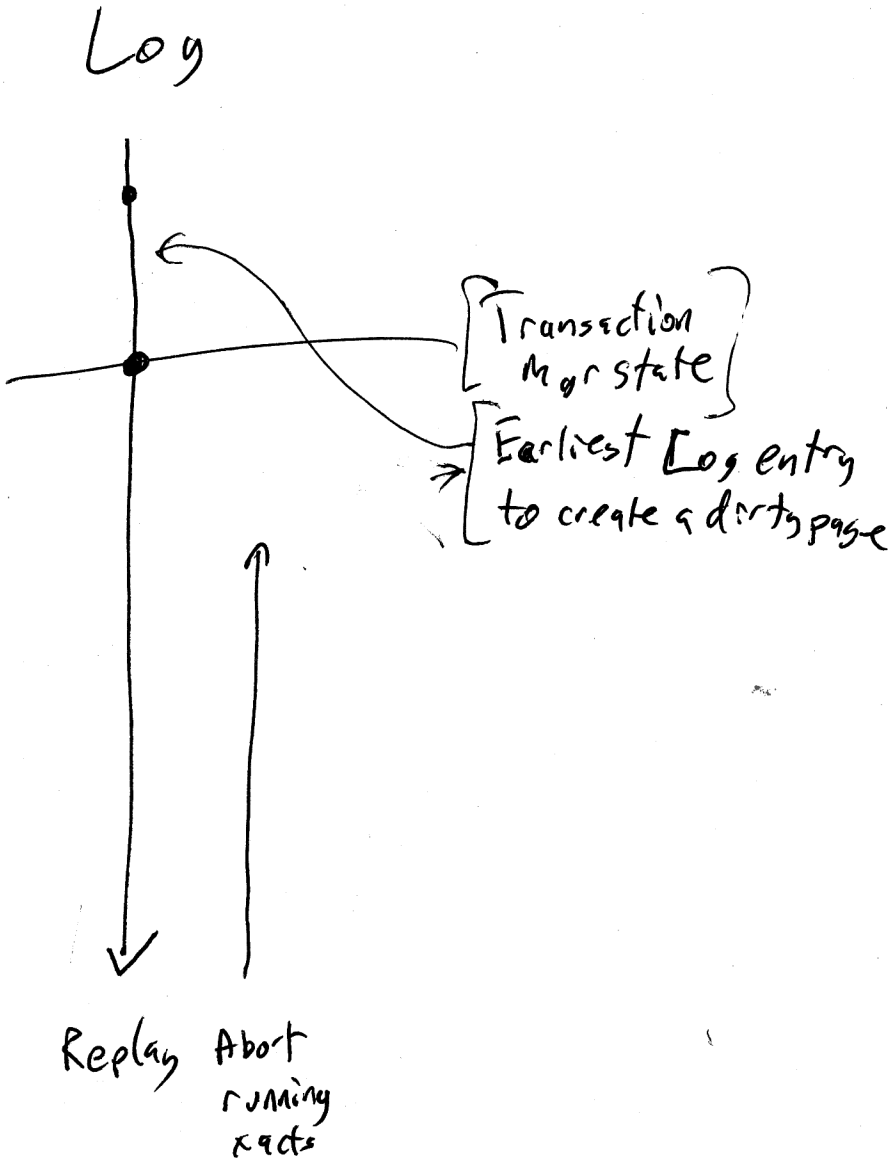
↓ ← DB_2 + changes




P0 P1 P2 P3 P4 P5
T1: P3
T1: commit
T2: P4=
T3: P0:
T2: Abort
T3: commit
Differential Files
Merged as of T3:
P0 P1 P2 P3 P4 P5


P0 P1 P2 P3 P4
T1 P2
T1 P3
T1 ABORT
T1 w(P1), w(P3)
T1 ABORT
time


Buffer Manager
(transient)
Disk
(persistent)
0 1 2 3 4 5
Log
(persistent)
Transaction Manager
T1 = running
T2 = running
Log: T1 W(4)
P4
Dirty
5
T1 Commit
T1 P4
T1 commit
T1 = committed
T2 = running
Once on disk we're good
P4
Dirty
T1 needs abort
T1 = running
T2 = commit
ok because T2 never wrote


Log
Transaction Mgr state
Earliest Log entry to create a dirty page
Replay
Abort running xacts