

GESTIONE DEL RIPRISTINO



CREDITS

- Paolo Ciaccia. Sistemi Informativi L-B
Home Page del corso:
 - <http://www-db.deis.unibo.it/courses/SIL-B/>

SOMMARIO

- Obiettivi gestione del ripristino
- Ripristino basato sui file di log
- Problemi prestazionali ripristino
- Ripristino per media failure



SOMMARIO

- **Obiettivi gestione del ripristino**
- Ripristino per transaction e system failure
- Problemi prestazionali ripristino
- Ripristino per media failure



GESTIONE DEL RIPRISTINO

- Sappiamo già che ogni transazione o termina con commit o con abort
- Dopo un abort o un commit non può più “cambiare idea”
- Anche in caso di **malfunzionamenti**:
 - Gli effetti delle transazioni **committed** devono essere permanenti (**Durability**)
 - Gli effetti delle transazioni **aborted** non devono lasciare tracce (**Atomicity**)
- Durability e Atomicity vengono assicurati da una particolare componente del Transaction Manager: il **gestore del ripristino**



MALFUNZIONAMENTI

Transaction failure

è il caso in cui una transazione abortisce per sua scelta (ROLLBACK)

gli effetti della transazione sul DB devono essere annullati

System failure

il sistema ha un guasto hardware o software che provoca l'interruzione di tutte le transazioni in esecuzione, senza però danneggiare la memoria permanente (dischi)

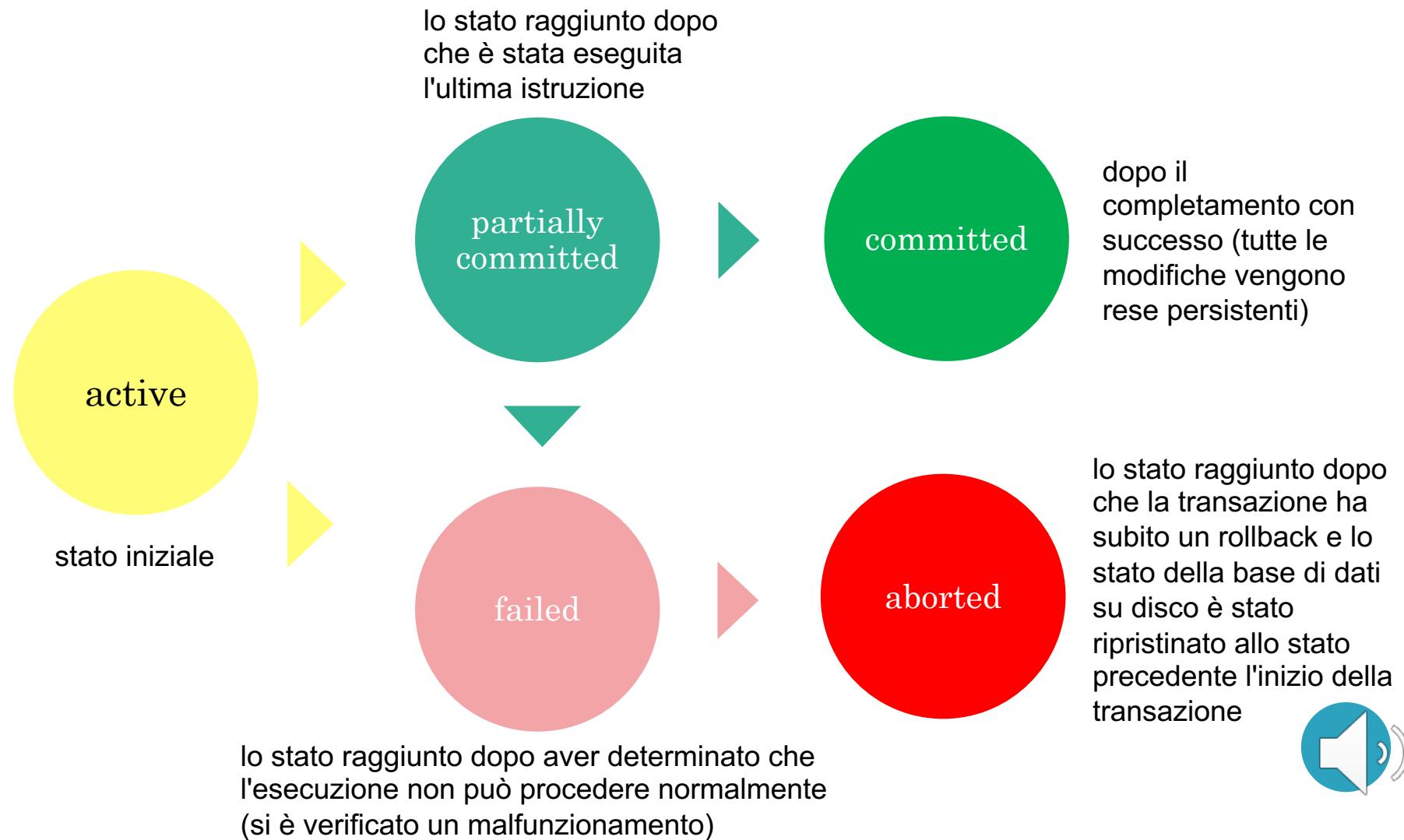
spesso dovuto a problemi a **memoria volatile** (memoria principale e cache)

Media failure

il contenuto (persistente) della base di dati viene danneggiato
problemi a **memoria non volatile** (dischi, nastri)



MODELLO ASTRATTO DI ESECUZIONE



MODELLO ASTRATTO DI ESECUZIONE - OSSERVAZIONI

- In caso di Abort, eventuali **scritture esterne osservabili** (cioè scritture che non possono essere "cancellate", ad es. su terminale o stampante) eseguite dalla transazioni **non possono essere eliminate**
- Dopo il rollback di una transazione, il sistema ha due possibilità:
 - **rieseguire la transazione**: ha senso solo se la transazione è stata abortita per **system failure** o **media failure**
 - **eliminare la transazione** se si verificano **transaction failures** che possono essere corretti solo riscrivendo il programma applicativo



ESEMPIO

T1	A	B
R(A)	50k	150k
$A = A - 10k$	50k	150k
W(A)	40k	150k
R(B)	40k	150k
$B = B + 10k$		
W(B)		
Commit		

Si riesegue T

T1	A	B
R(A)	40k	150k
$A = A - 10k$	40k	150k
W(A)	30k	150k
R(B)	30k	150k
$B = B + 10k$	30k	150k
W(B)	30k	160k
Commit	30k	160k

GUASTO

Non si riesegue T

40k	150k
-----	------

in entrambi i casi lo stato risultante è inconsistente
 → abbiamo modificato la base di dati prima di avere la certezza che la transazione avrebbe terminato con successo



SOMMARIO

- Obiettivi gestione del ripristino
- **Ripristino per transaction e system failures**
- Problemi prestazionali ripristino
- Ripristino per media failure



RECOVERY CON LOG

- Le attività di ripristino eseguite a valle del verificarsi di un **transaction o di un system failure** vengono genericamente indicate con il termine **ripresa a caldo**
- Durante l'esecuzione di una transazione tutte le operazioni di scrittura sono registrate in un **file di log, memorizzato su memoria stabile**

Memoria stabile

- teoricamente non è mai coinvolta in failures (astrazione teorica)
- se ne implementano approssimazioni, duplicando le informazioni in diverse memorie non volatili con probabilità di fallimento indipendenti

File di log

- nell'implementazione effettiva possono essere usati più file fisici
- scritture sequenziali: un record, con informazioni minimali, per ogni modifica di blocco/pagina eseguita dalla transazione + info inizio (BEGIN) e fine transazioni (COMMIT/ROLLBACK)
- Per semplicità nel seguito assumiamo che un blocco corrisponda a un singolo record



RECOVERY CON LOG

Se una pagina P del DB viene modificata dalla transazione T, il Log contiene un record del tipo **(LSN, T, PID, before(P), after(P), prevLSN)**

T= identificatore della transazione

LSN= Log Sequence Number
(n. progressivo del record nel Log)

PID= identificatore della pagina modificata

before(P) =
“before image”di P, ovvero il contenuto di P prima della modifica

after(P) = “after image”di P, ossia il contenuto di P dopo la modifica

LSN	T	PID	before(P)	after(P)	prevLSN
...					
235	T1	BEGIN			-
236	T2	BEGIN			-
237	T1	P15	(abc, 10)	(abc, 20)	235
238	T2	P18	(def, 13)	(ghf, 13)	236
239	T1	COMMIT			237
240	T2	P19	(def, 15)	(ghf, 15)	238
241	T3	BEGIN			-
242	T2	P19	(ghf, 15)	(ghf, 17)	240
243	T3	P15	(abc, 20)	(abc, 30)	241
244	T2	ROLLBACK			242
245	T3	COMMIT			243
...					

prevLSN = LSN del precedente record del Log relativo a T

record che specificano l'inizio (**BEGIN**) di una transazione e la sua terminazione (**COMMIT** o **ROLLBACK**)



PROTOCOLLO WAL - WRITE-AHEAD LOGGING

- Affinché il Log possa essere utilizzato per ripristinare lo stato del DB a fronte di malfunzionamenti, è importante che venga applicato il cosiddetto protocollo **WAL** (Write-ahead Logging)

prima di scrivere su disco una pagina P modificata, il corrispondente record di Log deve essere già stato scritto nel file di Log

- Se il protocollo WAL **non viene rispettato** è possibile che
 - una transazione T modifichi il DB aggiornando una pagina P
 - si verifichi un system failure prima che il Log record relativo alla modifica di P sia stato scritto nel Log
 - in questo caso non ci sarebbe alcun modo di riportare il DB allo stato iniziale → **si perde atomicità**
- La **persistenza** viene garantita quando la transazione ha terminato la sua esecuzione con successo (ha eseguito COMMIT) e tutti i record di log sono stati scritti su memoria stabile → la transazione può passare nello stato **committed**



La responsabilità di garantire il rispetto del protocollo WAL è del **Buffer Manager**, che gestisce il buffer del DB e il buffer del Log (maggiori dettagli in seguito)

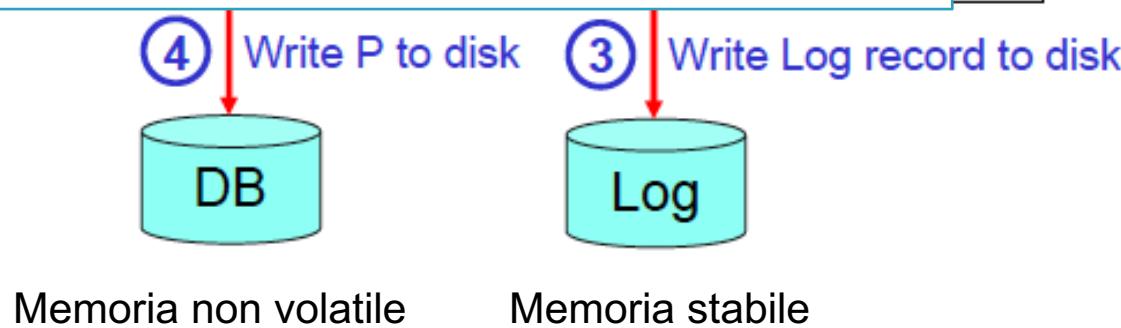
Stato corrente della base di dati
=
riflette tutte e sole le azioni delle transazioni committed
=
Stato corrente della base di dati su disco + Log su disco

Stato corrente della base di dati su disco

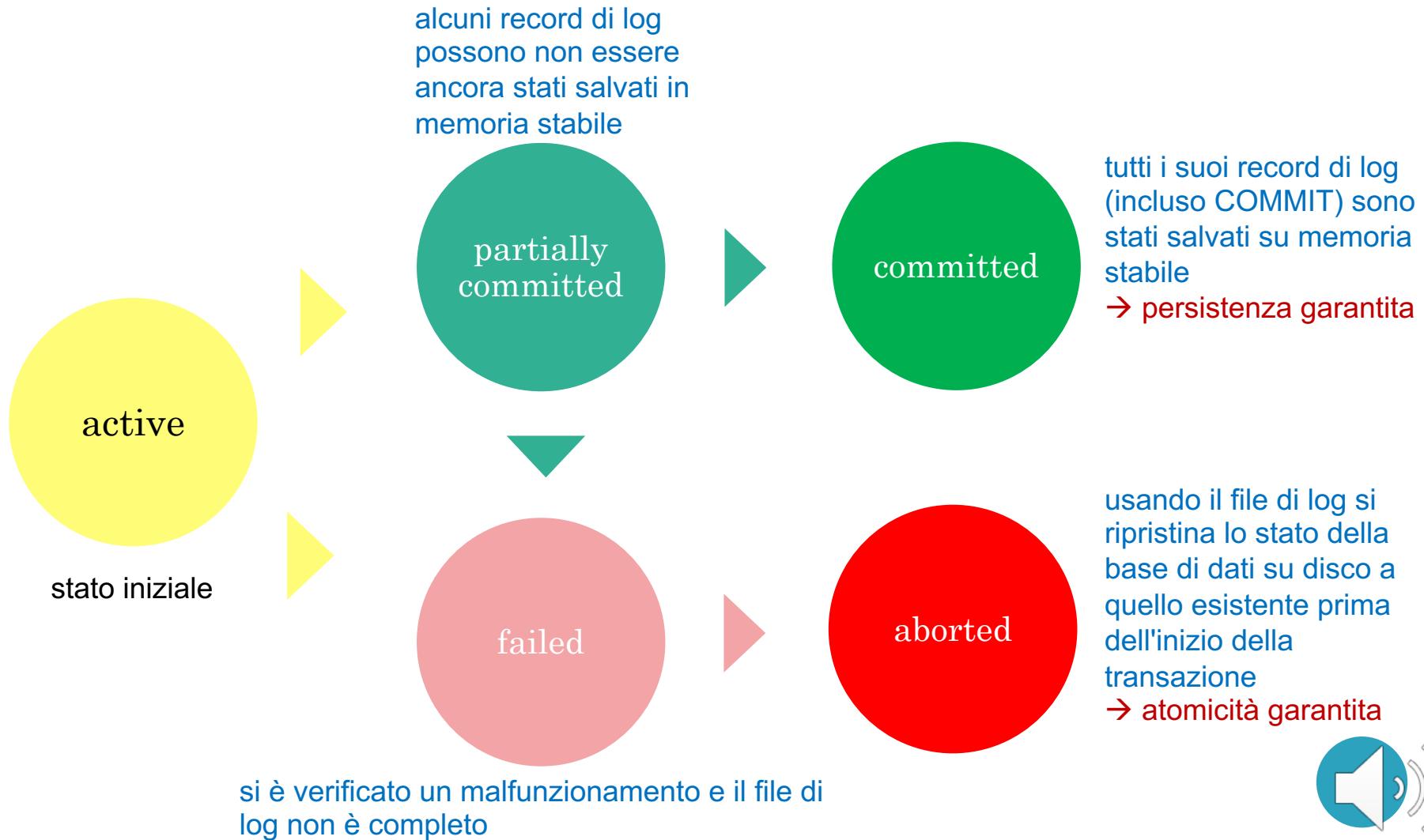
- riflette solo le azioni delle transazioni il cui effetto è già stato reso persistente
- può non riflettere alcune azioni di transazioni committed (alcuni blocchi devono ancora essere forzati su disco)
- può riflettere azioni di transazioni non committed (alcuni blocchi sono stati forzati su disco ma devono essere riportati allo stato iniziale)



Memoria volatile

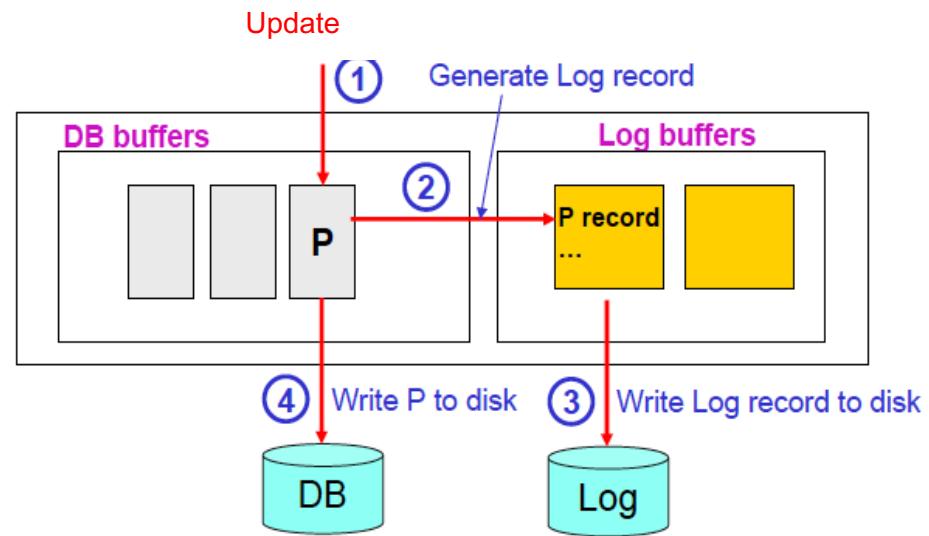


PROTOCOLLO WAL E MODELLO ASTRATTO DI ESECUZIONE



IMPLEMENTAZIONE DEL PROTOCOLLO WAL

- Passo 3 sempre eseguito prima del Passo 4
- Quando eseguire Passo 4 rispetto a **quando la pagina P viene modificata nel buffer?**
- Quando eseguire il Passo 4 rispetto a **quando la transazione entra nello stato committed?**



IMPLEMENTAZIONE DEL PROTOCOLLO WAL

Quando eseguire Passo 4 rispetto a
quando la pagina P viene modificata nel buffer?

Politica No-Steal

Si mantiene la pagina P nel buffer e si attende che T abbia eseguito COMMIT prima di scriverla su disco (copia su disco **dal momento del COMMIT, non prima**)

Peggiora gestione del buffer: si rischia di esaurire lo spazio a disposizione in memoria centrale

Politica Steal

Si scrive P su disco **quando più conviene** (per liberare il buffer o per ottimizzare le prestazioni di I/O), eventualmente anche prima della terminazione di T

Migliore gestione del buffer



IMPLEMENTAZIONE DEL PROTOCOLLO WAL

Quando eseguire il Passo 4 rispetto a
quando la transazione entra nello stato **committed**?

Politica Force

Prima di scrivere il record di COMMIT sul Log si forza la scrittura su disco di tutte le pagine modificate da T

Poi si scrive il record di COMMIT sul file di Log

Quando T entra nello stato **committed**, tutte le modifiche sono state rese persistenti su disco

Molti I/O inutili: se una pagina P è frequentemente modificata, deve essere scritta su disco ad ogni COMMIT

Politica No-Force

Si scrive subito il record di COMMIT sul file di Log

Le pagine del buffer si copiano su disco successivamente (appena si può)

Quando T entra nello stato **committed**, alcune delle sue modifiche possono ancora non essere state rese persistenti

Migliora gli accessi a disco: una pagina P viene scritta su disco solo se deve essere rimpiazzata nel buffer



TRANSACTION FAILURE

T esegue ROLLBACK (entra in stato failed)

Politica Steal

Alcune pagine modificate da una transazione potrebbero già essere state scritte su disco

UNDO di T

- Log a modifiche immediate
- si scandisce il Log a ritroso (usando i prevLSN) e si ripristinano nel DB le before(P) delle pagine P modificate da T

Politica No-Steal

Nessuna pagina modificata è già stata copiata su disco

Mai UNDO di T

- Inutile mantenere before(P) nel log
- Log a modifiche differite

LSN	T	PID	before(P)	after(P)	prevLSN
...					
236	T2	BEGIN			-
237	T1	P15	(abc, 10)	(abc, 20)	235
238	T2	P18	(def, 13) ← (ghf, 13)	236	
239	T1	COMMIT			237
240	T2	P19	(def, 15) ← (ghf, 15)	238	
241	T3	BEGIN			-
242	T2	P19	(ghf, 15) ← (ghf, 17)	240	
243	T3	P15	(abc, 20)	(abc, 30)	241
244	T2	ROLLBACK			242



SYSTEM FAILURE

Problema alla memoria centrale (ROLLBACK di sistema)

La transazione T è nello stato **committed**
(ha già scritto COMMIT nel file di log)

Tutte le modifiche sono state rese persistenti su disco

La transazione T non è nello stato **committed**

La transazione entra nello stato **failed**

Si procede come per **Transaction Failure**

Politica No-Force

Non è detto che tutte le modifiche operate da T siano state riportate su disco

REDO di T

→ si scandisce il Log a ritroso (usando i prevLSN) e si ripristinano nel DB le before(P) delle pagine P modificate da T

Politica Force

Tutte le modifiche operate da T sono già state riportate su disco

Mai REDO di T

LSN	T	PID	before(P)	after(P)	prevLSN
...					
235	T1	BEGIN			-
236	T2	BEGIN			-
237	T1	P15	(abc, 10) → (abc, 20)		235
238	T2	P18	(def, 13)	(ghf, 13)	236
239	T1	COMMIT			237
...					

ATTENZIONE

REDO e UNDO devono essere idempotenti

Più esecuzioni in sequenza devono essere equivalenti ad un'esecuzione singola

Si assicura un comportamento corretto anche in presenza di malfunzionamenti durante l'esecuzione della procedura di ripristino

- REDO da eseguire quando REDO è già in esecuzione
- UNDO da eseguire quando UNDO è già in esecuzione



SOMMARIO

- Obiettivi gestione del ripristino
- Ripristino per transaction e system failure
- **Problemi prestazionali ripristino**
- Ripristino per media failure



PROBLEMI PRESTAZIONALI

- La politica **Steal** è quella più utilizzata perché non richiede di mantenere nel buffer necessariamente tutti i blocchi modificati da transazioni che non hanno effettuato il commit
- La politica **Force** è quella meno utilizzata a causa dei costi
- Quindi la combinazione steal-no force è la più utilizzata dai DBMS
- Costoso se le transazioni sono tante e lunghe!
 - In caso di guasto, UNDO di transazioni **non committed**
 - Al COMMIT, copia su disco di tutte le pagine modificare dalle transazioni
- Costoso se le transazioni sono tante e lunghe!



CHECKPOINT

- Per migliorare le prestazioni, periodicamente si può eseguire un **checkpoint**, ovvero una scrittura forzata su disco delle pagine modificate
- Il sistema **periodicamente** (al checkpoint):
 1. forza tutte le **pagine di log** nel buffer su **memoria stabile**
 2. forza tutte le **pagine dati** nel buffer su **disco**
 3. forza il record <CKP> (o <checkpoint>) sul log in **memoria stabile**
- In caso di system failure, se T ha eseguito COMMIT prima del checkpoint, si è sicuri che per T non si dovrà eseguire REDO in caso di guasto

T1 e T2 non devono essere rifatti!

LSN	T	PID	before(P)	after(P)	prevLSN
237	T3	P15
238	T2	P18
239	T1	P17
240	T1	COMMIT			...
241	T2	COMMIT			...
242		CKP			
243	T3	P19
244	T3	COMMIT			...



SOMMARIO

- Obiettivi gestione del ripristino
- Ripristino per transaction e system failure
- Problemi prestazionali ripristino
- **Ripristino per media failure**



MEDIA FAILURE

Recovery con log

System Failure
Transaction Failure

Ripresa a caldo

Media Failure

Recovery con log + dump

Dump (backup): copia completa della base di dati, memorizzata in memoria stabile

La creazione del dump viene registrata nel file di log, con indicazione del file e del device su cui è stato effettuato il dump

Ripresa a freddo:

- si accede al dump e si ripristina il contenuto della base di dati su memoria non volatile
- si effettua una ripresa a caldo, accedendo al file di log come discusso in precedenza

