

Strutture fisiche e strutture di accesso ai dati (I parte)



ALBERTO BELUSSI

PRIMA PARTE

ANNO ACCADEMICO 2018-2019

DBMS e memoria secondaria

2

- Le basi di dati gestite da un DBMS risiedono in memoria secondaria, in quanto sono:
 - Grandi: non possono essere contenute in memoria centrale, e
 - Persistenti: hanno un tempo di vita che non è limitato all'esecuzione dei programmi che le utilizzano
- Caratteristiche della memoria secondaria:
 - Non è direttamente utilizzabile dai programmi
 - I dati sono organizzati in blocchi (o pagine)
 - Le uniche operazioni possibili sono la lettura e la scrittura di un intero blocco (pagina)
 - Il costo di tali operazioni è ordini di grandezza maggiore del costo per accedere ai dati in memoria centrale

Gestore del buffer

3

L'interazione tra
memoria secondaria e memoria centrale
avviene attraverso il
trasferimento di pagine della memoria secondaria
in una zona appositamente dedicata
della memoria centrale detta
BUFFER

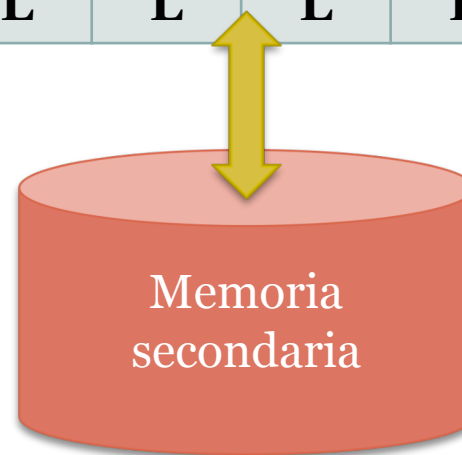
BUFFER

Si noti che:

- Il buffer è una zona di memoria condivisa dalle applicazioni
- Quando uno stesso dato viene utilizzato più volte in tempi ravvicinati il buffer evita l'accesso alla memoria secondaria
- La gestione ottimale del buffer è strategica per ottenere buone prestazioni nell'accesso ai dati

BUFFER

$B_{2,0}$	L	L	$B_{1,1}$	L	L
$B_{1,1}$	$B_{1,1}$	$B_{1,0}$	$B_{1,0}$	L	L
L	L	$B_{0,0}$	L	$B_{0,0}$	$B_{0,0}$
L	L	L	L	$B_{0,1}$	L
L	L	L	$B_{0,0}$	$B_{0,0}$	L
L	L	L	L	L	L



$B_{i,j}$ indica che nella pagina del buffer è caricato il blocco B , inoltre “i” indica che il blocco è attualmente utilizzato da i transazioni mentre “j” è 1 se il blocco è stato modificato e 0 altrimenti. **L** indica pagina libera.

Gestore del buffer

5

Il buffer è organizzato in PAGINE. Una pagina ha le dimensioni di un blocco della memoria secondaria.

Il gestore del buffer si occupa del caricamento/salvataggio delle pagine in memoria secondaria.

POLITICA del gestore dei buffer:

- In caso di richiesta di lettura di un blocco,
 - ✦ se il blocco è presente in una pagina del buffer allora non si esegue una lettura su memoria secondaria e si restituisce un puntatore alla pagina del buffer,
 - ✦ altrimenti si cerca una pagina libera e si carica il blocco nella pagina, restituendo il puntatore alla pagina stessa.

Gestore del buffer

6

POLITICA del gestore dei buffer:

- In caso di richiesta di scrittura di un blocco precedentemente caricato in una pagina del buffer, il gestore del buffer può decidere di differire la scrittura su memoria secondaria in un secondo momento.

In entrambi i casi (lettura e scrittura) l'obiettivo è quello di aumentare la velocità di accesso ai dati.

Tale comportamento del gestore dei buffer si basa sul principio di LOCALITA': "I dati referenziati di recente hanno maggiore probabilità di essere referenziati nuovamente in futuro". Inoltre, una nota legge empirica dice che: "il 20% dei dati è acceduto dall'80% delle applicazioni". Tutto ciò rende conveniente dilazionare la scrittura su memoria secondaria delle pagine del buffer.

Gestione delle pagine

7

Dati necessari per la gestione delle pagine:

- Per ogni pagina del buffer si memorizza il blocco contenuto indicando il file e il numero di blocco (o offset)
- Per ogni pagina del buffer si memorizza un insieme di variabili di stato tra cui si trovano sicuramente:
 - ✦ Un contatore per indicare il numero di transazioni che utilizzano le pagine
 - ✦ Un bit di stato per indicare se la pagina è stata modificata o no.

Gestione delle pagine

8

Primitive per la gestione delle pagine:

- Fix: viene usata dalle transazioni per richiedere l'accesso ad un blocco e restituisce al chiamante un puntatore alla pagina contenente il blocco richiesto. Passi della primitiva:
 - ✦ Il blocco richiesto viene cercato nelle pagine del buffer, in caso sia presente, si restituisce un puntatore alla pagina contenente il blocco
 - ✦ Altrimenti, viene scelta una pagina libera P (contatore = zero). Si sceglie la pagina in base a criteri diversi: la pagina usata meno di recente (LRU) o caricata da più tempo (FIFO). Se il bit di stato di P è a 1, P viene salvata in memoria secondaria (operazione di flush). Si carica il blocco in P aggiornando file e numero di blocco corrispondenti.
 - ✦ Se non esistono pagine libere, il gestore del buffer può adottare due politiche (STEAL): “ruba” una pagina ad un'altra transazione (operazione di flush); oppure (NO STEAL) sospende la transazione inserendola in una coda di attesa. Se una pagina si libera (contatore = zero) il gestore si comporterà come al punto precedente.

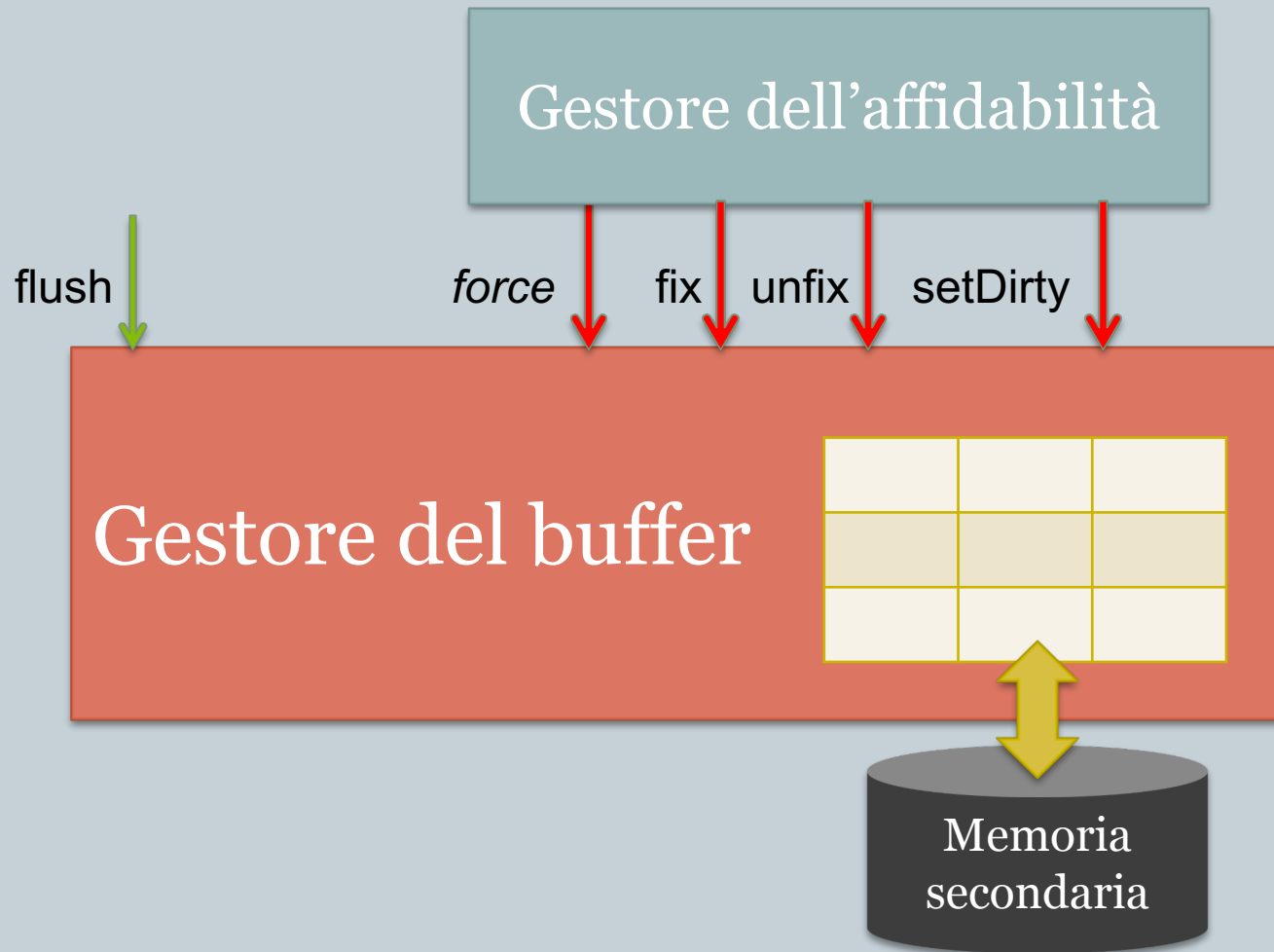
Gestione delle pagine

9

- ✦ Quando una transazione accede ad una pagina per la prima volta il contatore si incrementa.
- setDirty: viene usata dalle transazioni per indicare al gestore del buffer che il blocco della pagina è stato modificato
 - ✦ L'effetto è la modifica del bit di stato a 1
- unfix: viene usata dalle transazione per indicare che la transazione ha terminato di usare il blocco
 - ✦ L'effetto è il decremento del contatore di utilizzo della pagina.
- force: viene usata per salvare in memoria secondaria in modo SINCRONO il blocco contenuto nella pagina (primitiva usata dal modulo Gestore dell'Affidabilità).
 - ✦ L'effetto è il salvataggio in memoria secondaria del blocco e il bit di stato posto a zero.
- flush: viene usata dal gestore del buffer per salvare blocchi sulla memoria secondaria in modo ASINCRONO. Tale operazione libera pagine “dirty” (il bit di stato posto a zero)

Gestore del buffer

10



Gestore dell'affidabilità

11

E' il modulo responsabile di ciò che riguarda:

- L'esecuzione delle istruzioni per la gestione delle transazioni e
- La realizzazione le operazioni necessarie al ripristino della base di dati dopo eventuali malfunzionamenti

Per il suo funzionamento il gestore dell'affidabilità deve disporre di un dispositivo di MEMORIA STABILE.

MEMORIA STABILE = memoria resistente ai guasti

Gestore dell'affidabilità

12

In memoria stabile viene memorizzato il file di LOG che registra in modo sequenziale le operazioni eseguite dalle transazioni sulla base di dati.

RECORD di LOG:

○ Record di TRANSAZIONE:

- ✦ Begin della transazione T: record $B(T)$
- ✦ Commit della transazione T: record $C(T)$
- ✦ Abort della transazione T: record $A(T)$
- ✦ Insert, Delete e Update eseguiti dalla transazione T sull'oggetto O:
 - record $I(T,O,AS)$: $AS=$ After State
 - record $D(T,O,BS)$: $BS=$ Before State
 - record $U(T,O,BS,AS)$

Gestore dell'affidabilità

13

I record di transazione salvati nel LOG consentono di eseguire in caso di ripristino le seguenti operazioni:

- **UNDO:** per disfare un'azione su un oggetto O è sufficiente ricopiare in O il valore BS; l'insert/delete viene disfatto cancellando/inserendo O;
- **REDO:** per rifare un'azione su un oggetto O è sufficiente ricopiare in O il valore AS; l'insert/delete viene rifatto inserendo/cancellando O;

Gli inserimenti controllano sempre l'esistenza di O (non si inseriscono duplicati)

Proprietà di IDEMPOTENZA:

$\text{UNDO}(\text{UNDO}(A)) = \text{UNDO}(A)$

$\text{REDO}(\text{REDO}(A)) = \text{REDO}(A)$

Gestore dell'affidabilità

14

RECORD di LOG:

○ Record di SISTEMA:

- ✦ Operazione di **DUMP** della base di dati: record di DUMP
- ✦ Operazione di **CheckPoint**: record $CK(T_1, \dots, T_n)$ indica che all'esecuzione del CheckPoint le transazioni attive erano T_1, \dots, T_n .

Gestore dell'affidabilità

15

Operazione di CheckPoint: è un'operazione svolta periodicamente dal gestore dell'affidabilità; prevede i seguenti passi:

- Sospensione delle operazioni di scrittura, commit e abort delle transazioni
- Esecuzione di operazione di force sulle pagine “dirty” di transazioni che hanno eseguito il commit
- Scrittura sincrona sul file di LOG del record di CheckPoint con gli identificatori delle transazioni attive
- Ripresa delle operazioni di scrittura, commit e abort delle transazioni

Gestore dell'affidabilità

16

Regole per la scrittura sul LOG ed esecuzione delle transazioni

- Regola WAL (Write Ahead Log): i record di log devono essere scritti sul LOG prima dell'esecuzione delle corrispondenti operazioni sulla base di dati (garantisce la possibilità di fare sempre UNDO)
- Regola Commit-Precedenza: i record di log devono essere scritti sul LOG prima dell'esecuzione del COMMIT della transazione (garantisce la possibilità di fare sempre REDO)

Tali regole consentono inoltre di salvare i blocchi delle pagine “dirty” in modo totalmente asincrono rispetto al commit delle transazioni.

ESECUZIONE DEL COMMIT DI UNA TRANSAZIONE

Una transazione sceglie in modo atomico l'esito di COMMIT nel momento in cui scrive nel file di LOG in modo sincrono (primitiva force) il suo record di COMMIT.

Gestore dell'affidabilità

18

Operazioni di ripristino in caso di guasto

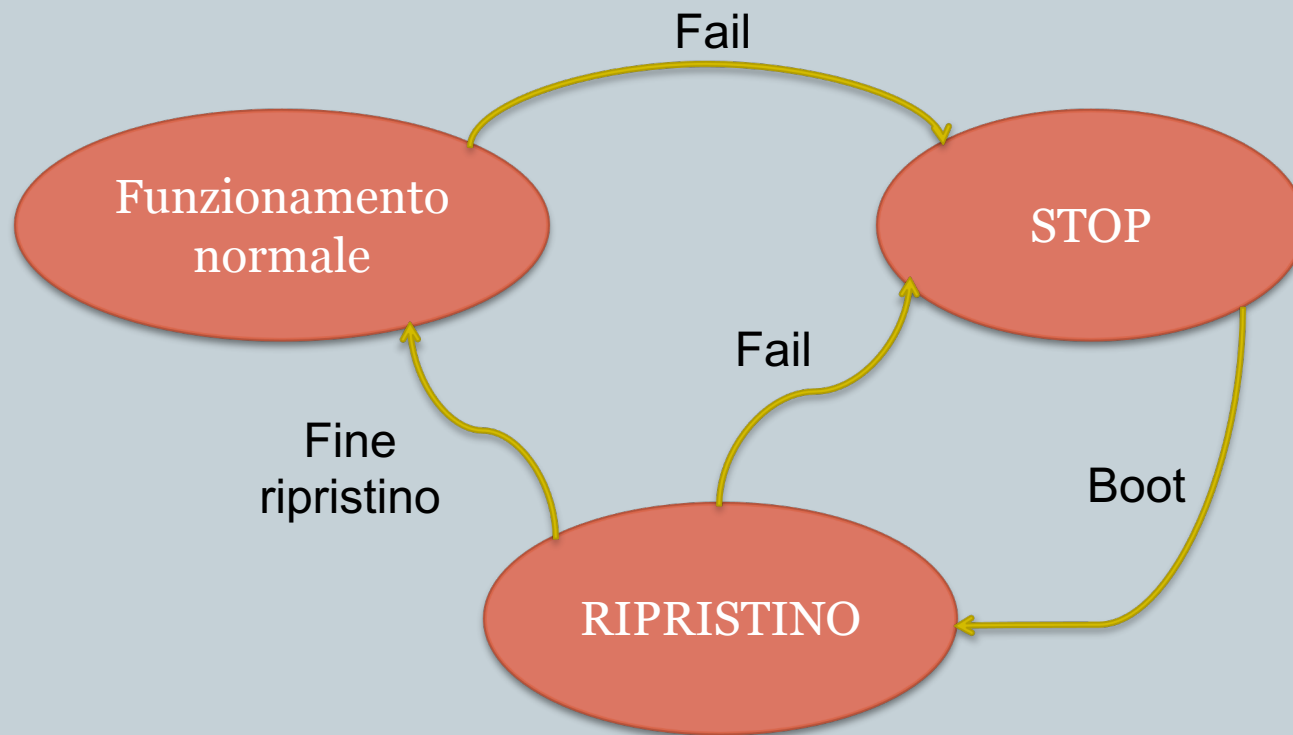
Tipi di guasto:

- Guasto di sistema: perdita del contenuto della memoria centrale
→ Ripresa a caldo
- Guasto di dispositivo: perdita di parte o tutto il contenuto della base di dati in memoria secondaria
→ Ripresa a freddo

Gestore dell'affidabilità

19

Modello di funzionamento



Ripresa a caldo

20

Passi:

- Si accede all'ultimo blocco del LOG e si ripercorre all'indietro il log fino al più recente record CK.
- Si decidono le transazioni da rifare/disfare inizializzando l'insieme UNDO con le transazioni attive al CK e l'insieme REDO con l'insieme vuoto.
- Si ripercorre in avanti il LOG e per ogni record B(T) incontrato si aggiunge T a UNDO e per ogni record C(T) incontrato si sposta T da UNDO a REDO.
- Si ripercorre all'indietro il LOG disfacendo le operazioni eseguite dalle transazioni in UNDO risalendo fino alla prima azione della transazione più vecchia.
- Si rifanno le operazioni delle transazioni dell'insieme REDO

Ripresa a freddo

21

Passi:

- Si accede al DUMP della base di dati e si ricopia selettivamente la parte deteriorata della base di dati
- Si accede al LOG risalendo al record di DUMP
- Si ripercorre in avanti il LOG rieseguendo tutte le operazioni relative alla parte deteriorata comprese le azioni di commit e abort
- Si applica una ripresa a caldo

Esercizio

22

Data la seguente situazione del file di LOG:

B(T1), B(T2), U(T2,O1,B1,A1), I(T1,O2,A2), B(T3),
C(T1), B(T4), U(T3,O2,B3,A3), U(T4,O3,B4,A4),
CK(T2,T3,T4), C(T4), B(T5), U(T3,O3,B5,A5),
U(T5,O4,B6,A6), D(T3,O5,B7), A(T3), C(T5),
I(T2,O6,A8) guasto

Che passi svolge la ripresa a caldo?