

Tecnologia delle basi di dati

Prova di autovalutazione — 3 giugno 2005

Domanda 1

Il check-point, nei vari DBMS, viene realizzato in due modi diversi:

1. in alcuni sistemi si prende nota delle transazioni attive e si rifiutano (momentaneamente) nuovi commit
2. in altri si inibisce l'avvio di nuove transazioni e si attende invece la conclusione (commit o abort) delle transazioni attive

Spiegare, intuitivamente, le differenze che ne conseguono sulla gestione delle riprese a caldo.

Domanda 2

Illustrare, brevemente, ma in modo ordinato, le differenze nelle tecniche di implementazione fra i livelli di isolamento **SERIALIZABLE** e **REPEATABLE READ**, facendo anche riferimento alle diverse strutture fisiche, primarie e secondarie, che possono essere coinvolte. Spiegare, quindi, perché le conseguenti differenze di prestazioni possono essere in alcuni casi enormi e in altri relativamente piccole.

Domanda 3

Per ragionare su alcuni concetti e tecniche di controllo di concorrenza, è utile includere negli schedule anche le operazioni di commit: ad esempio c_i potrebbe indicare il commit della transazione i . Un esempio di schedule potrebbe quindi essere: $w_1(x)r_2(x)c_2w_3(y)c_3w_1(y)c_1$. In tale contesto, una classe di schedule interessante è la seguente:

Uno schedule s è *commit order-preserving conflict serializable (COCSR)* quando, per ogni coppia di transazioni t_i e t_j che vanno in commit in s , se due operazioni $o_i(x)$ di t_i e $o_j(x)$ di t_j sono in conflitto e $o_i(x)$ precede $o_j(x)$ in s , allora c_i precede c_j in s .

Informalmente, per tutte le transazioni che vanno in commit, l'ordine delle operazioni in conflitto è coerente con l'ordine dei commit.

Dimostrare che la classe di schedule COCSR è propriamente contenuta nella classe CSR (facendo riferimento, per l'ipotesi di commit-proiezione, alle sole transazioni che vanno in commit); mostrare cioè che COCSR è contenuta in CSR e che non è vero il viceversa.

Domanda 4

Nel controllo di concorrenza basato su timestamp una transazione viene uccisa se essa, avendo un timestamp pari a ts , richiede una scrittura su un elemento x tale che $ts < WTM(x)$. Alcuni autori hanno notato che questa uccisione non è in effetti necessaria. Spiegare perché (commentando anche con riferimento alla risposta alla domanda precedente).

Domanda 5

Si supponga di disporre di una base di dati con i saldi dei conti correnti gestiti dalle varie agenzie di una banca e di dover eseguire su di essa l'interrogazione che calcola, per ciascuna agenzia, la somma dei saldi dei conti correnti. Indicare (con un breve commento, non più di tre righe) quale livello di isolamento (**READ UNCOMMITTED**, **READ COMMITTED**, **REPEATABLE READ** o **SERIALIZABLE**) si potrebbe scegliere in ciascuno dei seguenti casi (si supponga che, in generale, sia stato rilevato che, nel corso degli inserimenti e delle modifiche, vengono inseriti valori sbagliati anche di vari ordini di grandezza, che sono poi corretti prima del commit):

1. l'interrogazione è eseguita mentre vengono inseriti alcuni conti correnti (in ciascuna agenzia pochi rispetto a quelli già presenti); l'operazione ha la finalità di acquisire informazioni anche approssimate sugli andamenti complessivi
2. l'interrogazione è eseguita mentre vengono modificati i saldi di alcuni conti correnti (in ciascuna agenzia pochi rispetto a quelli già presenti); l'operazione ha la finalità di stilare una classifica delle agenzie, sulla base della somma dei saldi
3. l'interrogazione è eseguita mentre vengono modificati i saldi di tutti i conti correnti; l'operazione ha la finalità di acquisire informazioni anche approssimate sugli andamenti complessivi
4. l'interrogazione è eseguita in un momento in cui non ci sono aggiornamenti
5. l'interrogazione è eseguita mentre vengono inseriti alcuni conti correnti (in ciascuna agenzia pochi rispetto a quelli già presenti); l'operazione ha la finalità di stilare una classifica delle agenzie, sulla base della somma dei saldi