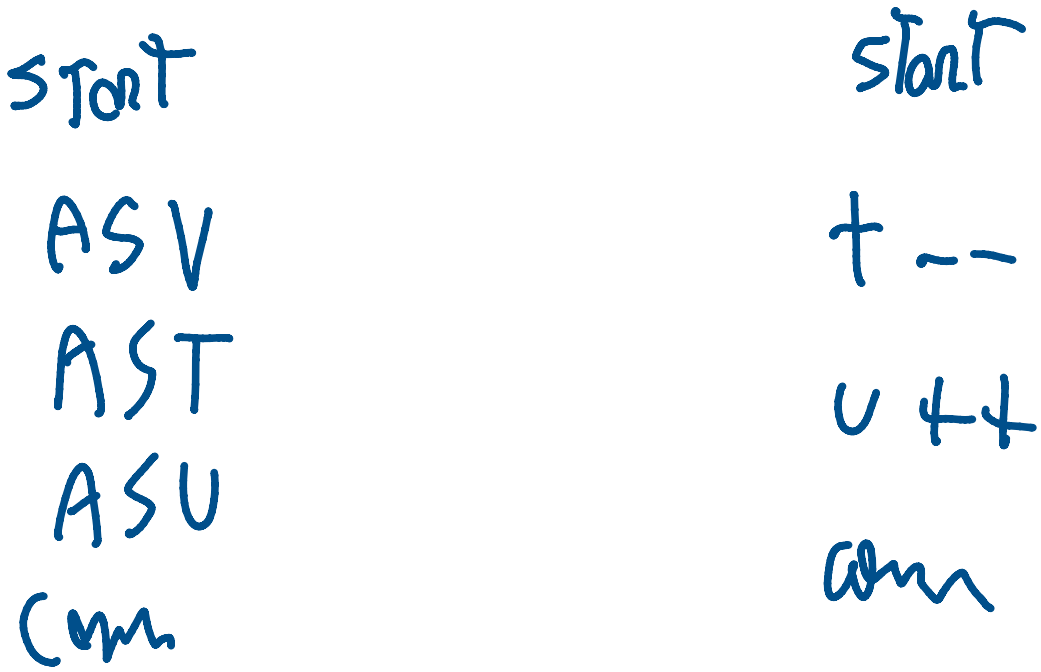
Esercizio orribile 20200702

domenica 14 febbraio 2021 15:45

lv 1 R.U.



Casi in cui va a buon fine senza anomalie:

- eseguo [t-- PRIMA di AST] AND [u++ PRIMA di ASU]
- eseguo [t-- DOPO di AST] AND [u++ DOPO di ASU]

(casi A e B)

Negli altri casi le transazioni terminano "con successo" ma riscontro l'anomalia dell'aggiornamento fansasma e ottengo statistiche sfalsante:

- Se esegue t-- dopo e u++ prima, ho statistiche sfalsate in ST (aggiungo una persona)
- Se eseguo t-- prima e u++ dopo, ho statistiche sfalsate in SU (perdo una persona)

(casi C e D)

lv = 2 RC

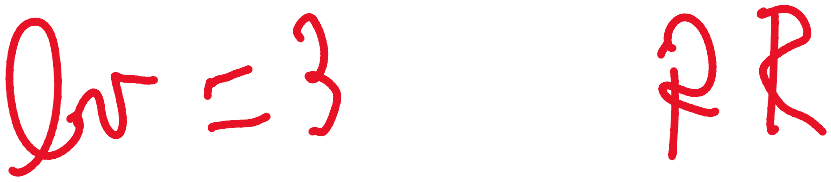
Nel lv 2 si applica, per le scritture 2PL (non stretto). Si considerano gli update operazioni atomiche di (lettura e) scrittura.

Se t-- e u++ vengono effettuate entrambe prima o dopo gli aggiornamenti delle statistiche, ovviamente non ci sono problemi. Anche se semplicemente t-- < AST e u++ < ASU non ci sono problemi (il 2PL non è stretto).

Se faccio t-- < AST e ASU < u++, la transazione t1 - mente fa ASU - viene messa in attesa (t2 non può rilasciare il lock per U++), quindi questa sequenza non è possibile (U++ viene eseguita PRIMA di ASU).

Se faccio invece AST < t-- < u++ < ASU, quando t2 richiede il lock per U++, viene messa in attesa, e si sbloccherà solo quando ASU è completata. Anche questo scheduling quindi non è possibile.

All'atto pratico, si osserva che con Read Committed, non si verifica nessuna anomalia.



Dal lv 3 in su (Repeatable Read e Serializable) si applica 2PL stretto, quindi:

Se t-- < u++ < AST < ASU oppure AST < ASU < t-- < u++ non ci sono problemi e le transazioni terminano senza attese e senza anomalie.

Come nel caso precedente, gli schedule problematici t-- < AST < ASU < u++ e AST < t-- < u++ < ASU non saranno possibili.

Inoltre, lo schedule t-- < AST < u++ < ASU non sarà neanche possibile, perché essendo applicato 2PL stretto, t2 non può anticipare il lock per u++.

In ogni caso la transazione termina con successo e senza anomalie.

In serializable, avendo il lock di predicato e essendo entrambi le transazioni utilizzatrici di Università, non potrò addirittura avere un'esecuzione concorrente, ma le 2 transazioni verranno eseguite in sequenza (si otterrà in pratica uno dei 2 possibili schedule seriali).

Ovviamente non ci saranno anomalie.