

Basi di dati II
Prof. Stefano Paraboschi
I prova - 31/10/2013

A. Si ha il seguente schedule:

$r_1(X), r_1(Y), r_2(Z), w_2(Z), r_1(Z), w_4(Y), r_2(Y), w_2(X), w_1(X), r_3(Y), w_3(X), r_5(X), w_5(Z)$

Classificare lo schedule rispetto alle classi VSR, CSR, 2PL, 2PL-strict, TS. In base alla classificazione, é possibile garantire che l'esecuzione sarà priva di anomalie o che saranno presenti anomalie?

B. Si ha un sistema distribuito con 3 nodi. Ciascun nodo ospita 2 pagine, ciascuna con 2 tuple. Nodo 1: $P_1(t_1, t_2), P_2(t_3, t_4)$; Nodo 2: $P_3(t_5, t_6), P_4(t_7, t_8)$; Nodo 3: $P_5(t_9, t_{10}), P_6(t_{11}, t_{12})$. Le transazioni nascono sul primo sistema su cui operano e sono attive su un solo nodo alla volta. Si assuma che le transazioni abbiano intenzione di eseguire le seguenti operazioni, nell'ordine specificato dallo schedule e che i lock vengano conservati fino alla fine dello schedule da tutte le transazioni.

$r_2(P_2), r_4(t_5), r_3(t_8), r_1(t_{10}), w_1(t_4), r_2(P_4), w_3(t_{10}), w_2(P_3), w_4(t_8)$

Mostrare le richieste di lock prodotte da ciascuna transazione e il grafo di attesa finale di ciascun nodo. Illustrare quindi l'applicazione dell'algoritmo per il rilevamento del deadlock distribuito.

C. Si ha un sistema con un buffer con capacità pari a 5. I primi tre elementi del buffer conservano le pagine P_{10}, P_{11} e P_{12} e sono attive. Il quarto elemento contiene una pagina P_{15} *dirty* e *unfixed*. Il quinto elemento contiene la pagina P_{20} *unfixed* e non *dirty*. Le transazioni in esecuzione chiedono quindi di accedere alle pagine P_{30}, P_{40} e P_{50} . Descrivere il comportamento del sistema nei due casi di politica *Steal* e *NoSteal*.