

Repetisjon av transaksjonshåndtering og samtidighetskontroll

Lana Vu
anhlv@ifi.uio.no

Repetisjon

- ACID-egenskapene
- Transaksjon
- Eksekveringsplan og serialiserbarhet
- Konfliktserialiserbarhet og presedensgraf
- 2-faselåsing (2 Phase Locking)
- Isolasjonsnivåer
- Snapshot isolation
- First Updater Wins (FUW)

ACID

Atomicity

- Alt eller ingenting. Hele transaksjonen feiler dersom en liten del av den feiler

Consistency

- Alle «krav/constraints» i databasen er oppfylt.

Isolation

- Transaksjoner skal ikke merke at andre transaksjoner utføres samtidig med dem selv

Durability

- Når transaksjoner er avsluttet, skal effekten av dem være varig og ikke kunne påvirkes av systemfeil

Transaksjon

En transaksjon er en *sekvens* av operasjoner som *bevarer konsistens* i databasen

Eksempler:

- Bankoverføringer
- Reservasjonssystemer
- Kioskautomaten i 3. etasje

Eksekveringsplan

En **eksekveringsplan** (schedule) S for en mengde transaksjoner $\{T_1, \dots, T_n\}$ er en **fletting** av operasjonene i T_1, \dots, T_n .

Hva er en «god» eksekveringsplan?

- En som garanterer *serialiserbarhet*

Serialiserbarhet

En **eksekvering** av en mengde **transaksjoner** er **seriell** hvis eksekveringen fullføres fullstendig for én transaksjon før neste transaksjon eksekveres.

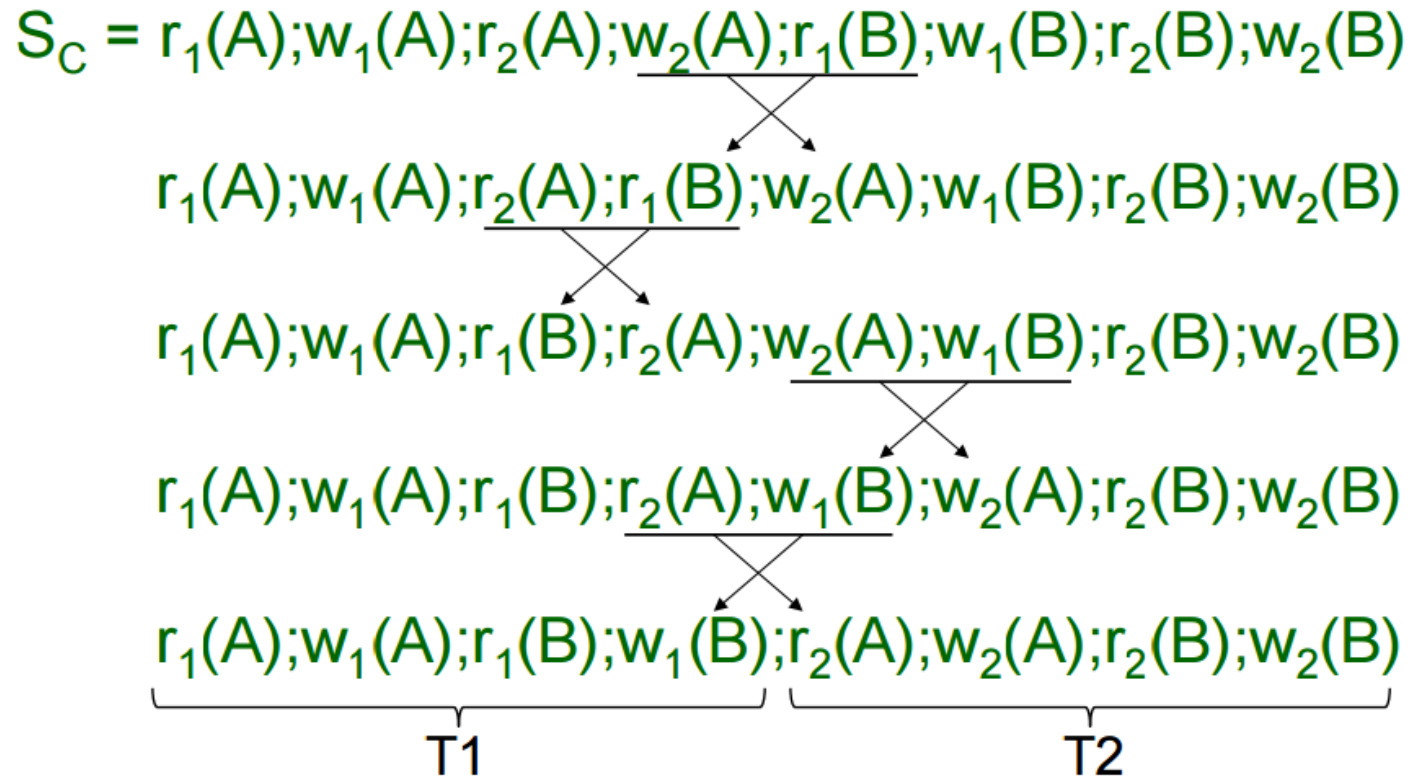
Eksekveringen er **serialiserbar** hvis transaksjonseksekveringene er slik at det fins en **seriell eksekvering** som gir samme totalresultat

Konfliktserialiserbarhet

To eksekveringsplaner S_1 og S_2 kalles *konfliktekvivalente* hvis S_1 kan omformes til S_2 ved en serie ombyttinger av naboooperasjoner som ikke er i konflikt med hverandre

En eksekveringsplan er **konfliktserialiserbar** hvis den er konfliktekvivalent med en *seriell eksekveringsplan*

Eksempel fra Ellen



S_C kan omformes til en seriell eksekveringsplan
Altså er S_C konfliktserialiserbar.



Presedensgraf

Kan finne ut om en eksekveringsplan er konfliktserialiserbar ved hjelp av en presedensgraf.

La S være en eksekveringsplan, og la $p_i(A)$ og $q_k(B)$ være to (vilkårlige) operasjoner i S . Notasjonen $p_i(A) <_S q_k(B)$ betyr at $p_i(A)$ gjøres før (kommer foran) $q_k(B)$ i S .

Hvordan sette opp presedensgrafene til en eksekveringsplan S :

- 1. Noder:** Transaksjonene i S
- 2. Kanter:** $T_i \rightarrow T_k$ (der $i \neq k$) dersom
 - $p_i(A) <_S q_k(A)$ og
 - minst en av p_i og q_k er en skriveoperasjon

Eksempel

Sykelfri graf indikerer konfliktserialiserbarhet

Tegn $P(S_1)$ for

$S_1 = r_1(A); r_2(A); r_3(B); w_1(A); r_2(C); r_2(B); w_2(B); w_1(C);$

Tegn $P(S_2)$ for

$S_2 = w_3(A); r_1(A); w_1(B); r_2(B); w_2(C); r_3(C)$



2-faselåsing

Kan ha shared locks og exclusive locks

Hvordan sikre serialiserbarhet?

Låser alene sikrer ikke dette. **2PL** er en låseprotokoll som garanterer serialiserbarhet.

Viktig regel

En transaksjon som har utført en unlock-operasjon, har ikke lov til å utføre flere lock-operasjoner.

Eksempel

Sett låser etter 2PL på S:

$S = r_1(A); r_2(A); r_3(B); w_1(A); r_2(C); r_2(B); w_2(B); w_1(C);$

$S = sl_1(A); r_1(A); sl_2(A); r_2(A); sl_3(B); r_3(B); u_3(B);$

$xl_1(A); w_1(A); sl_2(C); r_2(C); sl_2(B); r_2(B);$

$xl_2(B); w_2(B); u_2(A); u_2(B); u_2(C);$

$xl_1(C); w_1(C); u_1(A); u_1(C);$



Isolasjonsnivåer

Hvor isolert en transaksjon skal være fra andre transaksjoners endringer.

Full isolasjon: Seriell eksekveringsplan.

Høy isolasjon: Lav samtidighet og mye «overhead».

Lav isolasjon: Høy samtidighet, men det kan oppstå flere samtidighetsfenomener og –anomalier.

Vanlig å senke isolasjonsnivået for bedre ytelse. F.eks tillate dirty reads, fantomlesing osv. som kan føre til feil



Snapshot isolation

- Ofte brukt pga. bedre ytelse enn serialiserbarhet.
- I SI vil en transaksjon T i hele sin levetid lese de committede verdiene databasen hadde ved $TS(T)$ (starttiden til transaksjon T).
- T blir altså ikke påvirket av skriveoperasjoner som andre transaksjoner gjør etter $TS(T)$.
- **SI sikrer ikke serialiserbarhet!**
- Men SI gir høy grad av fletting og er standardstrategien for samtidighetskontroll i de fleste av dagens kommersielle DBMSer.

First Updater Wins (SI-protokoll)

Se Ellen sine slides for protokoll

Eksempel:

Se eksamen 2013