CONCURRENCY DAN RECOVERY CONTROL

Pertemuan 14

Pengertian *Concurrency Control*

- *Concurrency control* atau pengontrolan konkurensi merupakan salah satu fungsi dari DBMS.
- Pada DBMS terpusat multi-user dapat mengakses sejumlah transaksi pada waktu bersamaan.
- Penggunaan multi-user pada DBMS berpeluang terjadinya inkonsistensi basis data. Maka perlu adanya pengendalian persaingan eksekusi transaksi (*concurrency control*).

Transaksi yang konkuren banyak dipilih dibandingkan transaksi secara serial:

A. Idle time (waktu menganggur) kecil.

Aktivitas transaksi terbagi 2, yaitu:

- Aktivitas I / O, seperti pengaksesan disk, penulisan ke monitor.
- Aktivitas CPU, seperti proses perhitungan, pembandingan.

B. Response time (waktu tanggap) lebih baik.

Permasalahan yang terjadi dalam konkurensi :

- *Lost update* (modifikasi yg hilang)
- *Uncommited dependency* (ketergantungan yg tidak sukses/modifikasi sementara)
- *Inconsistent analysis (*analisa yang tidak konsisten*)*

Lost update (modifikasi yg hilang)

- Masalah operasi yang sukses dari seorang pengguna kemudian ditindih oleh operasi update

Waktu	T1	T2	X
t1		begin_transaction	100
t2	begin_transaction	read(x)	100
t3	read(x)	x=x+100	100
t4	x=x-10	write(x)	200
t5	write(x)	Commit	90
t6	Commit		90

Modifikasi yang dipakai

Hilang modifikasi

Ket : Modifikasi pada T2 akan hilang karena T1 akan memodifikasi tanpa memperhatikan modifikasi dari T2 pada waktu t3

Uncommited dependency (ketergantungan yg tidak sukses/modifikasi sementara)

- Masalah terjadi saat suatu transaksi membaca data dari transaksi lain yang belum dicommit

Waktu	T1	Tt2	X
t1		begin_transaction	100
t2		read(x)	100
t3		x=x=100	100
t4	begin_transaction	write(x)	200
t5	read(x)		200
t6	x=x-10	Rollback	190
t7	write(x)		190
t8	Commit		190

Modifikasi sementara

Jika setelah rollback membaca/
modifikasi data, maka akan membaca/memodifikasi data yang salah

Inconsistent analysis

- Masalah terjadi saat satu transaksi membaca beberapa nilai tetapi transaksi kedua pada waktu sama memodifikasi nilai tersebut

Waktu	T1	T2	X	Y	Z	Jml
t1		begin_transaction	100	50	25	
t2	begin_transaction	sum=0	100	50	25	0
t3	read(x)	read(x)	100	50	25	0
t4	x=x-10	sum=sum+x	100	50	25	100
t5	write(x)	read(y)	90	50	25	100
t6	read(z)	sum=sum+y	90	50	25	150
t7	z=z+10		90	50	25	150
t8	write(z)		90	50	35	150
t9	commit	read(z)	90	50	35	150
t10		sum=sum+z	90	50	35	185
t11		Commit	90	50	35	185

Teknik pengontrolan konkurensi :

- **Locking :** Apabila suatu transaksi mengakses suatu data maka suatu lock (kunci) dapat mencegah pengaksesan oleh transasksi lain.
- **Timestamping :**Timestamping merupakan suatu identitas waktu dimulainya suatu transasksi. Timestamping mengatur prioritas transaksi berdasarkan timestamp. Timestamp terkecil merupakan transaksi paling duluan, jika terjadi konflik transaksi direstart.
- **Optimistic :** Berasumsi bahwa konflik jarang terjadi sehingga proses tetap berjalan & pengecekan dilakukan pada saat transaksi sudah di-*commit*. Jika ada konflik transaksi direstart

Metode Locking

- Prosedur untuk mengontrol pengaksesan data secara konkuren.
- Apabila satu transaksi mengakses basis data, suatu lock (kunci) akan menolak pengaksesan transaksi lain untuk mencegah modifikasi yang tidak benar.

Jenis kunci yang digunakan pada metode locking

- Kunci READ/S (digunakan bersama)

  Jika transaksi mempunyai kunci read terhadap suatu data, maka dia dapat melakukan operasi read tetapi tidak dapat melakukan operasi update terhadap data tersebut.

- Kunci WRITE/X (eksklusif)

  Jika transaksi mempunyai kunci write terhadap suatu data, maka dia dapat melakukan operasi read maupun operasi update terhadap data tersebut.

- Matriks Locking

	S	X
S	True	False
X	False	False

Deadlock

- Suatu situasi dimana dua atau lebih transaksi masing-masing menunggu (wait) suatu kunci yg ditahan oleh transaksi lain, untuk dilepaskan.
- Teknik menangani deadlock
  - *Deadlock prevention*, DBMS mengamati transaksi apakah menimbulkan deadlock & tidak akan membiarkan deadlock terjadi
  - *Deadlock prevention and recovery*, DBMS membiarkan terjadi deadlock, mengenalinya lalu menanganinya.

Transaksi deadlock

Waktu	T1	T2
t1	Begin_transaction	
t2	Write_lock(X)	Begin_transaction
t3	Read(X)	Write_lock(Y)
t4	X=X-10	Read(Y)
t5	Write(X)	Y=Y+100
t6	Write_lock(Y)	Write(Y)
t7	Wait	Write_lock(X)
t8	Wait	Wait
t9	Wait	Wait
t10	…	Wait
t11	…	…

Timestamp

- Timestamp, merupakan suatu identifikasi unik dibuat DBMS yg mengindikasikan waktu mulai relatif dari suatu transaksi. Dengan waktu sistem atau penambahan pada kounter logik setiap waktu transaksi mulai. Suatu protokol yg menyusun transaksi2 secara global, dimana transaksi yg tertua, transaksi dg timestamp terkecil, mendapat prioritas utama dari konflik transaksi tsb.

Optimistic

- Berbasis pada asumsi bahwa pada lingkungan tertentu, jarang terjadi konflik, sehingga lebih efisien membiarkan transaksi dieksekusi. Kemudian pada saat akan di-commit, diidentifikasi apakah akan timbul konflik, jika ya maka transaksi harus di rollback atau di proses ulang.

Recovery Control

- Recovery basis data merupakan suatu proses penyimpanan kembali basis data pada keadaan yang benar sebelum terjadi kegagalan(*failure*).

Penyebab Kegagalan

- System Crash (Kerusakan Sistem)

  Akibat kesalahan pada perangkat keras atau lunak, menyebabkan kehilanhan memori utama
- Media Failure (kegagalan pada media)

  Media tidak dapat dibaca, menyebabkan kehilangan sebagian dari penyimpanan sekunder
- Application Software Error (kesalahan pada perangkat lunak)

  Kesalahan logika yang mengakses basis data menyebabkan satu atau lebih transaksi mengalami kegagalan
- Natural Physical Disasters (bencana fisik yang natural)

  Kebakaran, air bah, gempa
- Sabotase

  Kerusakan pada data, fasilitas perangkat lunak dan keras yang disengaja

Latihan

1. Selesaikan masalah dibawah ini dengan metode Locking pada concurency control untuk permasalahan Lost Update (Masalah Kehilangan Modifikasi).

   Diketahui T1, T2 : Transaksi 1, Transaksi2

   t : waktu

   x : 50 (nilai awal x)

   Transaksi yang terjadi :

t1 → T1 begin transaction	t10 → T2 write(x)
t2 → T1 write_lock(x)	t11 → T2 unlock(x)
t3 → T1 read(x)	t12 → T1 write_lock(x), T2 commit
t4 → T1 x=x+25, T2 begin transaction	t13 → T1 read(x)
t5 → T1 write(x), T2 write_lock(x)	t14 → T1 x=x-5
t6 → T1 unlock(x)	t15 → T1 write(x)
t7 → T1 Rollback	t16 → T1 unlock(x)
t8 → T2 read(x)	t17 → T1 commit
t9 → T2 x=x-25	

2. Selesaikan masalah dibawah ini dengan metode Locking pada concurency control untuk permasalahan Inconsistent Analysis (Analisa yang tidak konsisten).

Diketahui T1, T2 : Transaksi 1, Transaksi2

t : waktu

x : 200 (nilai awal x)

y : 150 (nilai awal y)

z : 75 (nilai awal z)

Transaksi yang terjadi :

t1 → T1 begin transaction	t12 → T2 commit
t2 → T1 sum=0, T2 begin transaction	t13 → T1 read(y)
t3 → T2 write_lock(y)	t14 → T1 sum=sum+y
t4 → T1 read_lock(y), T2 read(y)	t15 → T1 read_lock(z)
t5 → T2 y=y-20	t16 → T1 read(z)
t6 → T2 write(y)	t17 → T1 sum=sum+z
t7 → T2 write_lock(x)	t18 → T1 read_lock(y)
t8 → T2 read(x)	t19 → T1 read(y)
t9 → T2 x=x+15	t20 → T1 sum=sum+y
t10 → T2 write(x)	t21 → T1 unlock(y,z)
t11 → T2 unlock(x,y)	t22 → T1 commit