Obnovitev po zrušitvi

Iztok Savnik, FAMNIT

Prosojnice & učbenik

- Učbenik:
  - Raghu Ramakrishnan, Johannes Gehrke, *Database Management Systems, McGraw-Hill, 3rd ed., 2007.*
- *Prosojnice:*
  - *From „Cow Book“: R.Ramakrishnan, http://pages.cs.wisc.edu/~dbbook/*

Pregled: ACID lastnosti

- **A** tomicity: (atomarnost)

  Izvršijo se vse akcije Xact ali nobena.
- **C** onsistency: (konsistentnost)

  Če sta Xact in celotna PB konsistentna, potem je po transakciji PB konsistentna.
- **I** solation: (izolacija)

  Izvajanje ene Xact je izolirano od izvajanja drugih transakcij.
- **D** urability: (ohranjanje)

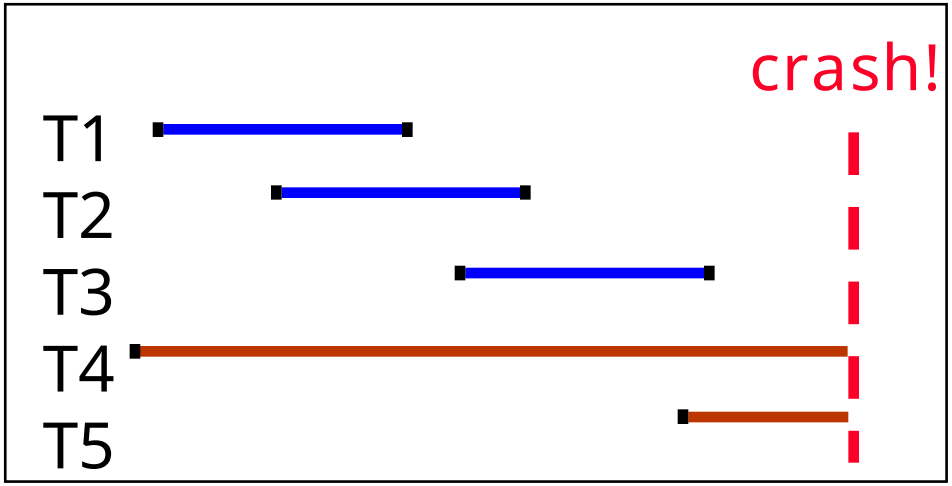
  Če Xact potrdi, so vse spremembe stalne.

- **Upravljalnik obnavljanja** garantira atomarnost & ohranjanje.

Motivacija

- Atomarnost:
  - Transakcija se lahko prekine (“Rollback”).
- Ohranjanje:
  - Kaj če neha delati SUPB? (Razlogi?)

- Pričakovano obnašanje po ponovnem zagonu sistema:
  - T1, T2 & T3 bi morale biti persistentne.
  - T4 & T5 bi morale biti prekinjene (izničene).



Predpostavke

- Kontrola vzporednosti se izvaja.
  - Striktni 2FZ, konkretno.
- Popravki se dogajajo „na mestu“.
  - Podatki so prepisani (izbrisani) na disk.

- Enostavna shema garantira atomarnost in ohranjanje.

Delo z izravnalnim bazenom strani

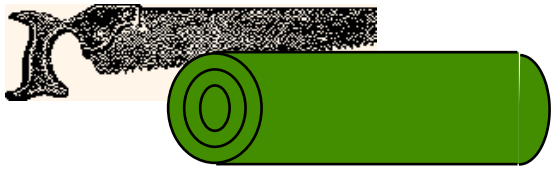
- Zagotovi vsa pisanja na disk?
  - Slab odzivni čas.
  - Omogoča ohranjanje podatkov.
- Ukradi okvirje izravnalnega bazena nepotrjenim Xacts?
  - Če ne, slab pretok.
  - Če da, kako zagotoviti atomarnost?

	No Steal	Steal
Force	Trivial	
No Force		Desired

Več o kraji in sili

- **KRAJA** (zakaj je doseganje atomarnosti težko)
  - *Ukrasti okvir F:* Trenutna stran v F (naj bo P) je zapisana na disk; neka Xact ima zaklep na P.
    - Kaj če Xact z zaklepom na P prekine izvajanje?
    - Moramo si zapomniti staro vrednost P ob kraji (za podporo UNDO pisanja na stran P).
- **BREZ SILE** (zakaj je doseganje ohranjanja težko)
  - Kaj če sistem pade preden se spremenjena stran zapiše na disk?
  - Zapiši čim manj je možno, na primerno mesto, ob potrjevanju, da bi omogočili REDO sprememb.

Osnovna ideja: Dnevniki



- Zapiši REDO in UNDO podatke za vsak popravek v dnevnik.
  - Sekvenčno pisanje v dnevnik (gre na ločen disk).
  - Minimalno podatkov (diff) zapisano v dnevnik; več popravkov gre na eno stran.
- Dnevnik: Urejen seznam REDO/UNDO akcij.
  - Zapis dnevnika vsebuje:

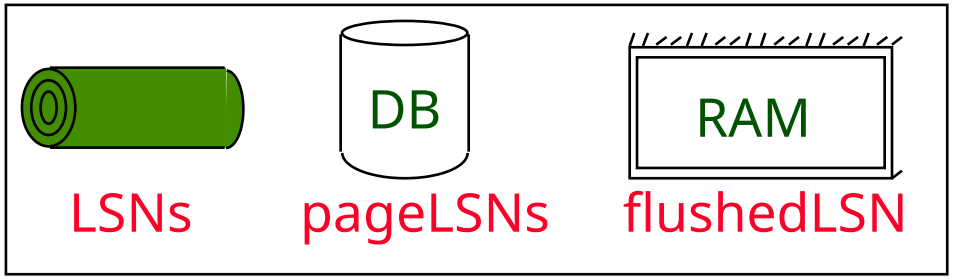
    <XID, pageID, offset, length, old data, new data>
  - in dodatne kontrolne podatke (do katerih pridemo kmalu).

Pisanje dnevnika vnaprej (WAL)

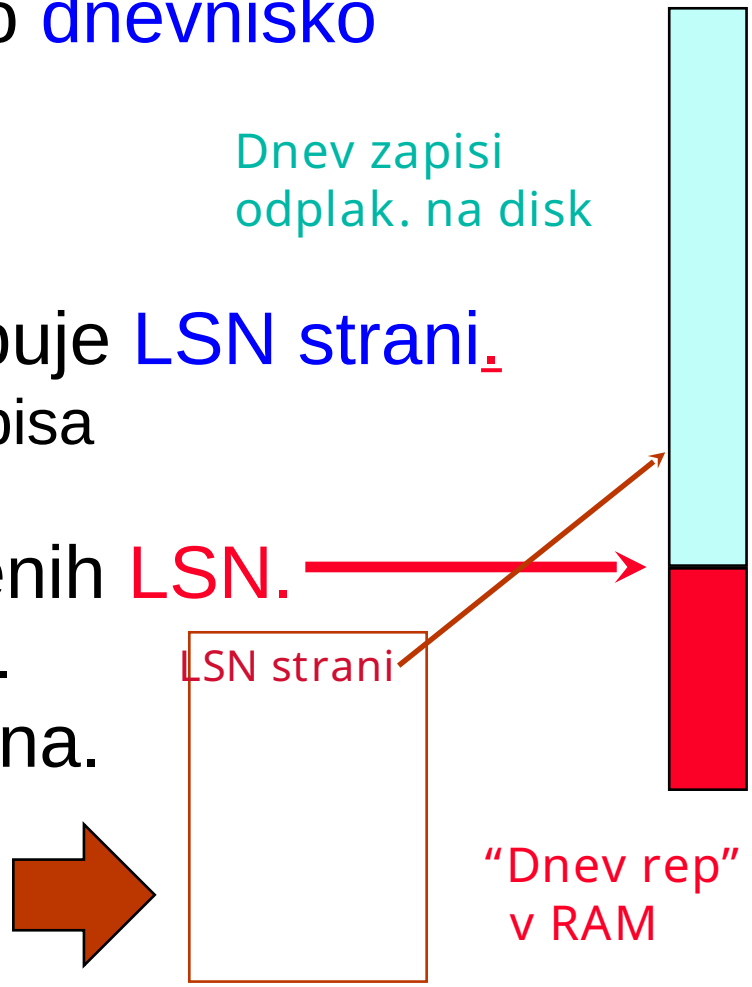
- Protokol Pisanje-dnevnika-vnaprej:
  - #1 Mora shraniti dnevniški zapis preden se pripadajoča stran zapiše na disk.
  - #2 Mora zapisati vse dnevniške zapise za Xact pred potrditvijo.
- #1 garantira atomarnost.
- #2 garantira ohranitev.

- Kako natančno je dnevnik (in obnovitev) narejen?
  - Študirali bomo ARIES algoritme.

WAL & dnevnik



- Vsak dnev zapis ima unikatno dnevniško sekvenčno številko.
  - Log Sequence Number (LSN).
  - LSN vedno narašča.
- Vsaka podatkovna stran vsebuje LSN strani.
  - LSN zadnjega dnevniškega zapisa popravka dane strani.
- Sistem hrani stanje odplaknjenih LSN.
  - Maks LSN odplaknjen do sedaj.
- WAL: Preden je stran zapisana.
  - page LSN $\leq$ flushed LSN



Dnevniški zapisi

Polja dnev zapisa:

prevLSN
XID
type
pageID
length
offset
before-image
after-image

samo za update zapise

Možni tipi dnev zapisov:

- Update
- Commit
- Abort
- End
  - konec potrjevanja ali prekinitve
- Kompenzacijski dnev zapisi (CLRs)
  - za UNDO akcije

Strukture povezane z dnevnikom

- Transakcijska tabela
  - En vpis za eno aktivno Xact.
  - Vsebuje XID, status (running/commited/aborted) in lastLSN.
- Tabela umazanih strani:
  - En vpis za eno umazano stran v izravnalnem bazenu.
  - Vsebuje recLSN -- LSN dnev zapisa, ki je prvi povzročil, da je stran umazana.

Normalno izvajanje Xact

- Vrsta branj & pisanj, ki jim sledi potrditev ali prekinitev.
  - Predpostavili bomo, da je pisanje na disk atomično.
    - V praksi imamo več podrobnosti okoli atomičnosti pisanja.
- Striktno 2FZ.
- Delo izravnalnika: STEAL, NO-FORCE + WAL dnevnik.

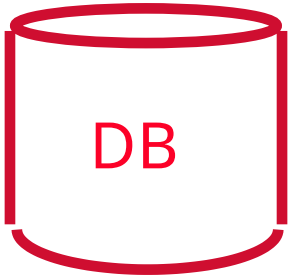
Kontrolna točka

- Periodično SUPB kreira kontrolno točko za minimizacijo časa porabljenega za obnovitev v primeru sistemske zrušitve. Zapiši v dnevnik:
  - begin_checkpoint zapis: Določa začetek KT.
  - end_checkpoint zapis: Vsebuje trenutno tabelo *Xact* in tabelo umazanih strani. To je mehka kontrolna točka':
    - Ostale Xacts nadaljujejo z delom; tabele so torej zanesljive samo v času begin_checkpoint zapisa.
    - Ne poskuša se na silo zapisati strani na disk; učinkovitost kontrolne točke je omejena z najstarejšimi nezapisanimi spremembami umazanih strani. (Dobro je ciklično zapisovati in odplakniti umazane strani na disk!)
  - Shrani LSN zapisa kontrolne točke na varno mesto (*master* zapis).

Ptičja perspektiva: Kaj je shranjeno kje

LogRecords

prevLSN
XID
type
pageID
length
offset
before-image
after-image

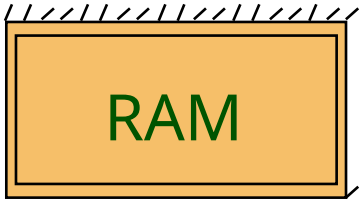


Data pages

each
with a
pageLSN

master record

LNS-KT



Xact Table

lastLSN
status

Dirty Page Table

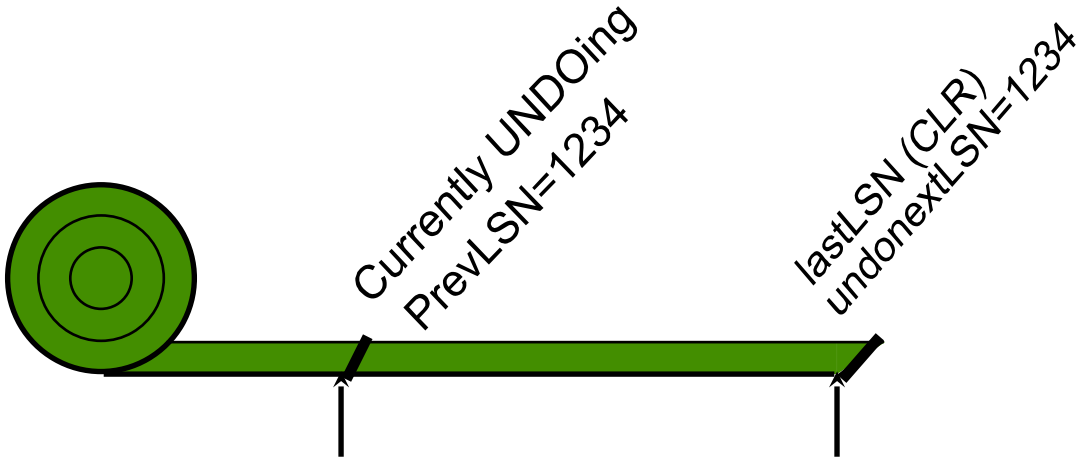
recLSN

flushedLSN

Enostavna prekinitev transakcije

- Najprej si poglejmo samo eksplic. prekinitev Xact.
  - Ni zrušitve sistema.
- Želimo “odvrteti nazaj” dnevnik v obratnem vrstnem redu z izničenjem (UNDO) sprememb.
  - Dobi lastLSN transakcije Xact iz tabele Xact.
  - Lahko sledimo dnev zapisom z uporabo polja prevLSN.
  - Pred začetkom vračanja z UNDO, zapiši dnev. zapis prekinitve (Abort)
    - Za reševanje iz zrušitve med vračanjem.

Prekini (2)

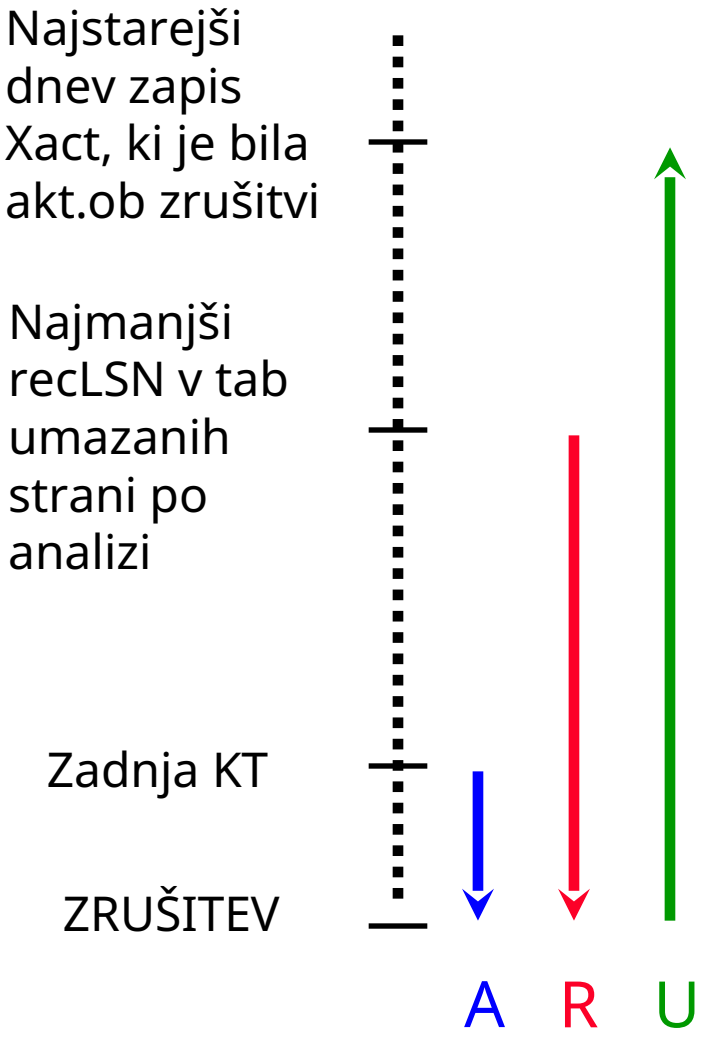


- Izvajanje UNDO zahteva zaklep na podatkih!
  - Ni problem!
- Preden restavriramo staro vrednost strani, se napiše CLR:
  - Dnevnik se lahko piše med UNDO!!
  - CLR ima eno dodatno polje: undonextLSN
    - Kaže na naslednji LSN za undo.
    - prevLSN zapisa, ki ga trenutno restavriramo.
  - CLR ni nikoli restavriran; Lahko je ponovno izveden pri ponavljanju zgodovine: zagotavlja atomarnst!
- Na koncu UNDO zapiši “end” dnevniški zapis.

Potrditev transakcije

- Zapiši potrdi (Commit) zapis v dnevnik.
- Vsi dnev. zapisi vse do lastLSN Xact so odplaknjeni.
  - Garantira flushedLSN >= lastLSN.
  - Odplakovanje dnevniških strani je sekvenčno, sinhrono zapisovanje na disk.
  - Veliko dnevniških zapisov na stran.
- Commit() se konča.
- Zapiši end zapis v dnevnik.

Obnovitev po zrušivi: celotna slika



- Začni od kontrolne točke (poiščemo jo preko master zapisa).
- Tri faze. Potrebno je:
  - Odkriti katera Xact je potrdila od kontrolne točke naprej in katera ni uspela (Analiza).
  - REDO vse akcije.
    - (ponovi zgodovino)
  - UNDO učinki neuspelih Xacts.

Obnovitev: faza analize

- Rekonstruiraj stanje ob kontrolni točki.
  - Uporabi end_checkpoint zapis.
- Preglej dnevnik naprej od kontrolne točke:
  - End zapis: Odstrani Xact iz Xact tabele.
  - Ostali zapisi: Dodaj Xact k Xact tabeli, postavi lastLSN=LSN, spremeni Xact status na potrdi.
  - Update zapis: Če P ni v Tabeli umazanih strani,
    - Dodaj P k T.U.S., postavi recLSN=LSN.

Obnovitev: Faza REDO

- Ponovimo zgodovino zato, da rekonstruiramo stanje ob zrušitvi:
  - Ponovno ovrednoti vse popravke (tudi za prekinjene Xact!), ponovno ovrednoti CLR.
- Pregled naprej od dnev zapisa, ki vsebuje najmanši recLSN v TUS.
- Za vsako CLR ali dnev zapis popravka LSN, naredi REDO akcije razen v primeru:
  - Spremenjena stran ni v TUS ali
  - Spremenjena stran je v TUS, vendar recLSN > LSN ali
  - pageLSN (v DB) $\geq$ LSN.
- REDO akcije:
  - Ponovi akcijo v dnevniku.
  - Postavi pageLSN na LSN. Ni dodatnega zapisa v dnev!

Obnovitev: Faza UNDO

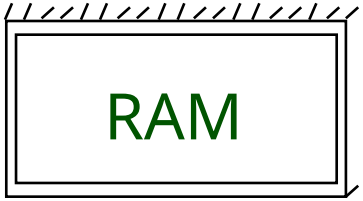
ToUndo={ / | / -- lastLSN prekinjenih Xact}

Repeat:

- Izberi največji LSN med ToUndo.
- Če je LSN CLR in velja undonextLSN==NULL
  - Zapiši End zapis za to Xact.
- Če je LSN CLR, in undonextLSN != NULL
  - Dodaj undonextLSN k ToUndo
- Sicer je LSN popravek. Izniči popravek, zapiši CLR in dodaj prevLSN k ToUndo.

Until ToUndo je prazen.

Primer obnovitve



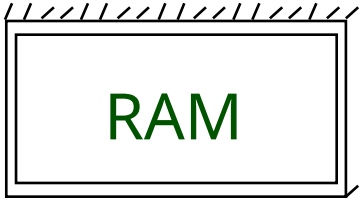
Xact Table
lastLSN
status
Dirty Page Table
recLSN
flushedLSN

ToUndo

LSN	LOG
00	begin_checkpoint
05	end_checkpoint
10	update: T1 writes P5
20	update: T2 writes P3
30	T1 abort
40	CLR: Undo T1 LSN 10
45	T1 End
50	update: T3 writes P1
60	update: T2 writes P5
	CRASH, RESTART

prevLSNs

Primer: Zrušitev med ponovnim zagonom



Xact Table
lastLSN
status
Dirty Page Table
recLSN
flushedLSN

ToUndo

LSN	LOG
00,05	begin_checkpoint, end_checkpoint
10	update: T1 writes P5
20	update T2 writes P3
30	T1 abort
40,45	CLR: Undo T1 LSN 10, T1 End
50	update: T3 writes P1
60	update: T2 writes P5
	CRASH, RESTART
70	CLR: Undo T2 LSN 60
80,85	CLR: Undo T3 LSN 50, T3 end
	CRASH, RESTART
90	CLR: Undo T2 LSN 20, T2 end

undonextLSN

Drugi problemi pri zrušitvi

- Kaj se zgodi, če se sistem zruši med analizo pri obnovi? Med fazo REDO?
- Kako omejiti količino dela med REDO?
  - Splakuj asinhrono v ozadju.
  - Pazi na “špice”!
- Kako omejiti količino dela pri UNDO?
  - Izogibaj se dolgo-trajajočim Xacts.

Pregled dnevnikov/obnove

- Upravljalnik obnove zagotavlja atomarnost & ohranjanje.
- Uporabi WAL skupaj z STEAL/NO-FORCE brez da bi žrtvovali korektnost.
- LSN identificira dnevniške zapise; povezani so v vzvratno verigo za vsako transakcijo (preko prevLSN).
- pageLSN omogoča primerjavo podatkovne strani in dnevniških zapisov.

Pregled dnevnikov/obnove

- Kontrolne točke: Enostaven način omejitve količine dnevniški zapisov pri obnovi.
- Obnova deluje v 3 fazah:
  - Analiza: Naprej od kontrolne točke.
  - Redo: Naprej od najstarejše recLSN.
  - Undo: Vzvratno od konca dnevnika do prve LSN najstarejše transakcije živeče ob zrušitvi.
- Med Undo, piši CLR zapise.
- Redo “ponovi zgodovino”: Poenostavi proces!