

Tosiakajärjestelmät – Luento 11: Tosiakatietokannat

Tiina Niklander

Perustuu Jan Lindströmin materiaalin S2002 ja artikkeliin:
Ramamritham, Son & DiPippo: *Real-Time Databases and
Data Services*. Real-Time Systems, 28, 179-215, 2004.

Sisältö

- n Abstrakti tietokantamalli
 - n (kertausta mm. tietokannan hallinta ja tapahtumankäsittely –kursseilta)
 - n Ajoitukset ja sarjallisuus
- n Aika ja eheys
- n Samanaikaisuudenhallinta
- n Esimerkkejä

Abstrakti tietokantamalli

- n Tietokanta on joukko nimettyjä tietoalkioita, joilla on arvo. Näiden arvot muodostavat tietokannan tilan.
- n Tietokantajärjestelmä on laite- ja ohjelmistokomponentit, jotka tarjoavat komennot tietokannan käsittelyyn eli nk. tietokantaoperaatiot.
- n Tietokantaoperaatiot suoritetaan atomisesti yksi kerrallaan eli sarjallisesti.
- n Tietokantajärjestelmä tukee myös transaktio-operaatioita: Begin (b), Read (r[x]), Write (w[x]), Commit (c) ja Abort (a)

Tietoalkio

- n d:(arvo, RTS, WTS, ...)
 - n arvo: tietoalkio tila
 - n RTS: viimeisin lukuaikaleima
 - n WTS: viimeisin kirjoitusaikaleima
- n Tietoalkiolla voi olla lisäinformaatiota, esim
 - n avi: absoluuttinen eheysaikaväli
 - n rvi: suhteellinen eheysaikaväli

Transaktion tilat

- n Transaktio on *sitoutunut* jos se on suorittanut commit operaation.
- n Transaktio on *peruuntunut* jos se on suorittanut abort operaation.
- n Transaktio on *aktiivinen* jos se on suorittanut begin operaation mutta ei commit tai abort operaatiota.
- n Transaktio ei voi olla vain yhdessä näistä tiloista.

ACID

- n Atomisuus: Transaktio suoritetaan kokonaan tai ei ollenkaan
- n Eheys: Transaktio siirtää tietokannan yhdestä eheestä tilasta toiseen.
- n Eristyneisyys: Transaktiot toimivat kuten olisivat yksin järjestelmässä.
- n Pysyvyys: Sitoutumisen jälkeen muutokset säilyvät virhetilanteissakin.

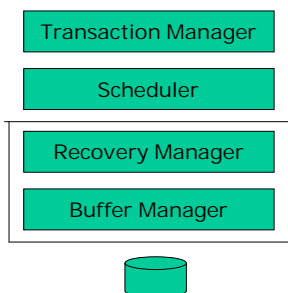
Ajoituksista

- n Oletetaan että transaktio yksinään muuttaa tietokannan eheästä tilasta eheään tilaan.
- n Transaktioiden suoritus voi olla sarjallinen eli kukin transaktio suoritetaan alusta loppuun peräjälkeen.
- n Sarjallinen ajoitus säilyttää tietokannan eheyden.
- n Sarjallistuva ajoitus säilyttää tietokannan eheyden.
- n Transaktioiden operaatiot muodostavat historian

Historian ominaisuuksia

- n Täydellinen historia on *sarjallistuva* jos se on ekvivalentti jonkin sarjallisen historian kanssa.
- n Historia on *ketjuperuutukset estävä* (cascading aborts) jos transaktio lukee vain sitoutuneiden transaktioiden kirjoittamia tietoalkioita.
- n Historia on *toipuva* (recoverable) jos transaktio sitoutuu vasta kun kaikki ne transaktiot ovat sitoutuneet, jolta transaktio on lukenut tietoalkion arvon.

Abstrakti tietokantamalli



Aika tietokannoissa

- n Temporaalitietokannat (temporal databases)
 - n datalla aikarajoitteita voimassaololle
- n Aktiiviset tietokannat (active databases)
 - n datalla voi olla aikarajoitteita
 - n osa tiedoista johdetaan (*derive*) automaattisesti muiden perusteella
- n Tosi-aikatietokannat (real-time databases)
 - n datalla ja transaktioilla aikarajoitteita
- n Mitattu data vs. johdettu data

Tosi-aikatietokannan eheys

- n Transaktioiden aikarajat (deadlines)
 - n tulokset saadaan aikarajojen puitteissa
- n Temporaalinen eheys (temporal concistency)
 - n tieto ajallisesti eheää absoluuttisesti ja suhteellisesti
- n Tietokannan eheys (sarjallistuvuus)
 - n tieto on loogisesti eheää

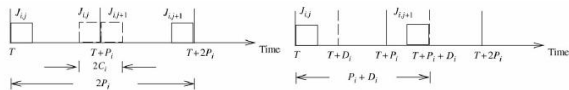
Temporaalinen eheys

- Absoluuttinen eheys (absolute consistency)
 - tieto eheää vain tietyn aikavälin
 - $\text{current time} - d.WTS \leq d.avi$
- Suhteellinen eheys (relative consistency)
 - tietoalkiojoukon oltava riittävän läheltä samaa ajanhetkeä. $\forall d, d' \in R: d.WTS - d'.WTS \leq R.rvi$
- Esim painekattilan mitta-arvot
 - Lämpö 100, Paine 360 OK
 - Lämpö 700, Paine 360 EI

Aika	Lämpö	Paine
100	100	360
200	300	720
300	700	100

Lyhytkestoisen datan päivitys

- n Säännöllisesti (esim. jaksollinen transaktio)
- n Kolme erilaista tapaa:
 - n Jakson pituus (P_i) on voimassaoloaika α_i
 - n Jakson pituus $\leq \alpha_i/2$, missä α_i voimassaoloaika
 - n Jakson pituus $> \alpha_i/2$ ja $P_i + D_i = \alpha_i$.



Tosi-aikatransaktion ominaisuudet

- n Aikaraja: deadline(T_i)
 - n hot, firm tai soft
- n Prioriteetti: priority(T_i)
- n Kriittisyys: criticality(T_i)

- n Kovat aikarajat - A priori takuu suoritukselle
 - n Tiedettävä ennalta käytettävät tietoalkiot ja WCET
- n Suurin osa tutkimuksesta kohdistuu pehmeisiin aikarajoihin

Samanaikaisuudenhallinnan ongelmat

- n Turha peruutus: transaktio peruuttaa toisen transaktion ja myöhemmin peruutuu itse.
- n Turha odotus: alempi prioriteettinen transaktio odottaa korkeamman päättymistä ja myöhemmin myöhästyy.
- n Turha suoritus: alempi prioriteettinen peruutetaan validoinnissa, koska konfliktissa korkeamman kanssa.
- n Turha uudelleenkäynnistys: transaktio peruutetaan vaikka historia on sarjallistuva.

Ajoitusmenetelmien luokittelu

- n Pessimistiset menetelmät
 - n Aikaleimamenetelmät
 - n Sarjallistuvuusverkkomenetelmät
 - n Lukitusmenetelmät
- n Optimistiset menetelmät
 - n Eteenpäin validoivat
 - n Taaksepäin validoivat
 - n Sarjallistuvuusverkkomenetelmät
- n Hybridit menetelmät

2PL-PA (Priority Abort)

- n 2PL + jos lukon varaaja on alemmpiprioriteettinen, se peruutetaan
- n Hyödyt:
 - n Korkeampi prioriteettisia suositaan
- n Haitat
 - n Ketjuperuutukset
 - n Lukkiuma
 - n Prioriteetin kääntyminen
- n Kutsutaan myös 2PL-HP (High Priority)

2PL-PI (Priority Inheritance)

- n Transaktio voi liittyä lukijoiden joukkoon vain jos sen prioriteetti suurempi kuin kenenkään kirjoittamista odottavan.
- n Lukon hallinnan ajan transaktion prioriteetti on vähintään suurimman odottajan prioriteetti.
- n Kutsutaan myös 2PL-WP (Wait Promote)

Optimistiset menetelmät

1. Lukuvaihe: Suoritetaan transaktion koodia ja luetut tietoalkiot lisätään lukujoukkoon ja kirjoitetut kirjoitusjoukkoon. Kirjoitukset yleensä tehdään vain paikallisiin tietoalkioihin.
2. Validointivaihe: Tarkistetaan että muodostettu historia on sarjallistuva, toipuva ja ketjuperuutukset estävä.
3. Kirjoitusvaihe: Muutokset kirjoitetaan osaksi tietokantaa.

OCC-BC (Broadcast Commit)

- n Validoiva transaktio lähettää viestin konfliktoiville transaktioille ja ne peruutetaan.
- n Pitkäkestoiset transaktiot perutaan herkästi.
- n Ei aikatietoinen (tai prioriteettitietoinen)

OPT-SACRIFICE

- n Jos validoinnissa yksikin konfliktoiva transaktio on korkeampi prioriteettinen kuin validoiva, validoiva peruutetaan.
- n Muuten konfliktoiva peruutetaan.
- n Ongelmana lähes valmiiden transaktioiden peruutuskustannus.

OPT-WAIT

- n Jos validoinnissa havaitaan yksikin korkeampi prioriteettinen konfliktoiva transaktio, validoitava jää odottamaan.
- n Odotuksen aikana validoiva voidaan joutua peruuttamaan.
- n Kuinka kauan odotetaan ???

WAIT-50

- n Validoiva transaktio odottaa kunnes vähemmän kuin puolella konfliktoivista transaktioista on korkeampi prioriteetti.
- n Tämän jälkeen konfliktoivat transaktiot peruutetaan.
- n Ongelmana peruutukset ja odotuksen pituus.

Validointimenetelmiä

- n Taaksepäin validointi: Konfliktit havaitaan tarkistamalla viimeaikoina sitoutuneiden transaktioiden kirjoitusjoukot ja validoivan transaktion lukujoukko.
- n Eteenpäin validointi: Konfliktit havaitaan tarkistamalla validoivan transaktion kirjoitusjoukko ja aktiivisten transaktioiden lukujoukko.

Intervallimenetelmä (OCC-TI)

- n Jokaisella transaktiolla T_i on aikaväli (timestamp interval) $TI(T_i)$.
- n Aluksi aikaväli on $[0, \infty[$, peruutetaan jos \emptyset
- n Aikaväli kuvaa niitä ajanhetkiä, jolloin transaktio voi sitoutua siten että historia on sarjallistuva.
- n Validoinnin aikana muutetaan aikavälejä dynaamisesti kuvaamaan transaktioiden järjestystä historiassa.

Prioriteettitietoisuus (priority cognizant)

- n Suorituksen tulisi suosia korkeampi prioriteettisia transaktioita.
- n Vaihtoehtoja:
 - n Jokaisessa konfliktissa
 - n Vain jos joudutaan peruuttamaan
- n Ei mitenkään selvää että kannattaa !

Kannattaako ?

- n 2PL
 - n Tulokset osoittaneet että prioriteettitietoisuus on hyödyksi.
- n Optimistiset
 - n Tulosten perusteella ei suurta vaikutusta kokonaistehokkuuteen (vaikutusta tärkeimpien läpikäymiseen)
- n Sarjallistuvuusverkoajoittajat
 - n Ei tuloksia

Toipuminen (recovery)

- n Pysyvä data
 - n Toipuminen abortoidusta transaktiosta edellyttää kantaan jo kirjoitettujen muutosten poistamista
- n Lyhytkestoinen data
 - n Peruuttaminen voidaan korvata mitätöinnillä, seuraava päivitys on kohta tulossa
- n Kannan osittaminen (hot ja cold)
 - n Pienemmät palautettavat yksiköt
 - n Erilainen käytötapa erilaiselle datalle

Tosiaikatietokannan palvelutaso

- n Tavanomaisia mittareita
 - n Deadline miss ratio – kuinka suuri osa transaktioista myöhästyi
 - n Data freshness – Kuinka suuri osa ajallaan sitoutuneiden transaktioiden käyttämästä lyhytkestoisesta datasta oli ajan tasalla
- n Lisäksi voidaan käyttää
 - n Overshoot – pahimman mahdollisen tilanteen toiminta (esim. myöhästymisten osuus ylikuormitustilanteessa)
 - n Settling time – kuinka kauan järjestelmän rauhoittuminen yo. tilasta kestää

QoS ja satunnainen ylikuormitus (transient overload)

- n Satunnainen ylikuormitus on lyhytkestoinen tilanne, jossa järjestelmässä on hetkellisesti enemmän töitä suoritettavana kuin mihin resurssit riittävät
- n Tällöin palvelun laatu aina kärsii
- n Tietyn palvelutason ylläpitäminen edellyttää
 - n Satunnaisten ylikuormitusten välttämistä
 - n Ylikuormitustilanteessa joustoa
 - n Tapahtumien takarajat
 - n Datan voimassaoloajat
 - n Tapahtuman keston lyhentäminen (epätarkempi tulos)

