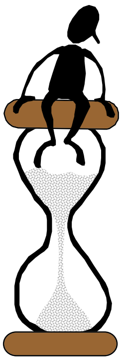


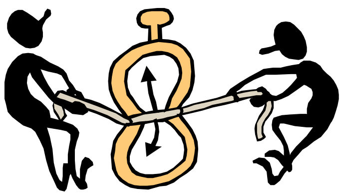
# Tosiaikatietokannat

Tiina Niklander

Tosiaikajärjestelmät seminaari 5.2.2002

## Sisältö

- Sovellusalueita
- Aikarajoitteet
  - Data
  - Transaktio
- Skedulointi ja rinnakkaisuudenhallinta
- Keskusmuisti vai levy
- Vikasietoisuus ja saatavuus



# Tosiaika ja tietokannat

| | oikeellisuus | ennustettavuus | tavoite | skedulointi |
|---|---|---|---|---|
| tosiaika-järjestelmät | aika (määräaika) | ennustettava tiedon käyttö ja suoritusaika | maksimoi aikarajoitettu suorituskyky | CPU |
| tietokannat | tietoeheys | ennustamaton datan käyttö | minimoi vasteaika | I/O |
| tosiaika-tietokannat (hajautetut) | aika- ja tietoeheys | ennustettava CPU ja ennustamaton datan käyttö | uhraa määräajat tiedon eheydelle | sekä CPU että I/O |

Son&Park: Scheduling Transactions for Distributed Time-Critical Applications.
In Readings in Distributed Computing Systems, 1994 s.592-618

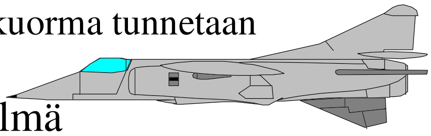
# Sovellusalueita 1

- Lennonjohtojärjestelmä
  - Useita lennonjohtopisteitä: kentät, aluelennonjohto
  - Kannassa jopa 20000 alkiota:
    - Lentokoneiden id, transponderikoodi, korkeus, lähtö- ja kohdekentät, reitti ja selvitykset
    - Kentän arkkitehtuuri, lentopinnat ja –reitit, lentorajoitukset,
  - Korkeintaan1 sekunnin viive mittauksesta ruudulle, kannalle sallitaan maks. 5 ms vasteaika.

Locke, Real-Time Databases: Real-World Requirements. Teoksessa Bestavros, Lin & Son (eds), Real-Time Database Systems: Issues and Applications. Kluwer, 1997. s.83-91

## Sovellusalueita 2

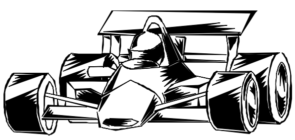
- Lentokoneen hallintajärjestelmä (sotilaskone)
  - Kannassa jopa 3000 alkiota: muita koneita 2000, maakohteita 250, karttoja, lentosuunnitelmia yms.
  - Kannan maksimivasteaika kork 1 ms, jotta järjestelmän maksimivaste alle 25 ms.
  - Aina keskusmuistikanta, kuorma tunnetaan etukäteen

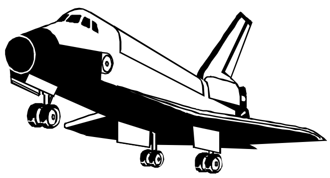


- Satelliitin hallintajärjestelmä
  - Vastaava kuin lentokoneen, mutta
  - Monennettu luotettavuuden lisäämiseksi

## Sovellusalueita 3

- Simulaattorit ja niiden ohjaus
  - Usein kaksi erillistä kantaa
    - Pysyvän ympäristön kuvaus: maisemat, kohteet, yms
    - Muuttuvan ympäriston kuvaus: mittarilukemat, liikkuvat laitteet, yms.
  - Todenmukaisen tilanteen luomiseksi kannalle sallitaan n. 5 ms viive.





## Sovellusalueita 4

- Prosessin ohjausjärjestelmä
  - Koko vaihtelee yhden venttiilin ohjauksesta kokonaiseen tuotannonohjausjärjestelmään
  - Vasteaikavaatimukset vaihtelevat
    - Muutaman millisekunnin vaste venttiilien ohjauksessa
    - Minuuttien vaste materiaalivirtojen ohjauksessa
  - Tyypilliset tietokantavaatimukset ovat 250-750 ms, jolloin koko järjestelmän vaste on 3-10 sekuntia. Nopeammissa toiminnoissa ei yleensä käytetä tietokantaa

## Väärinkäsityksiä

- Laitteistojen nopeus
- Tietokantatekniikat
- Tosiaika = nopeus

- Tosiaikavaatimukset perinteiseen kantaan
- Keskusmuistitietokanta = tosiaikatietokanta

- Tosiaikakannan ennustettavuus ei onnistu
- Tosiaikakanta on aina erikoiskanta (ei yleisesti käytettävissä)

## Datan eheys

- Looginen eheys
  - eheysrajoitukset

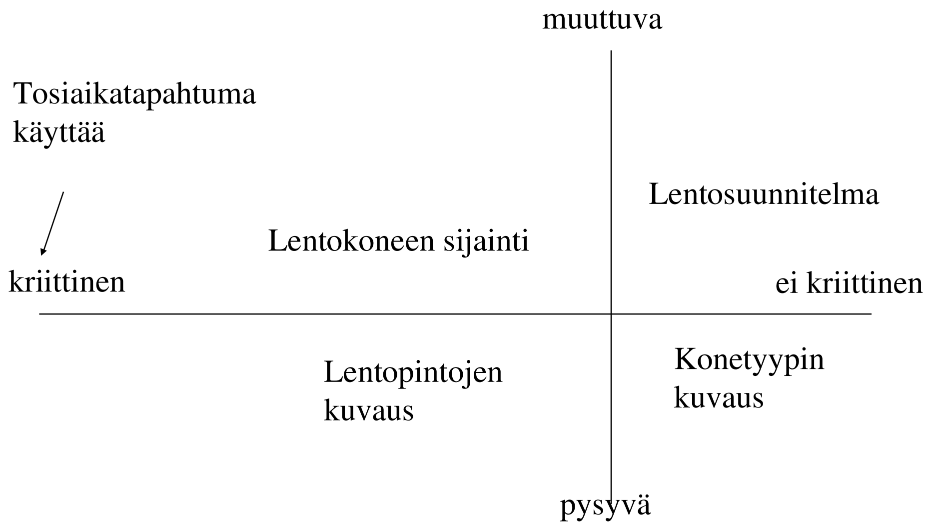
d:(arvo, abskesto, aika)
$d \in R$, R:(suhtkesto)

- Aikaeheys
  - Absoluuttinen aikaeheys
    - alkioiden arvojen suhde ympäristön tilaan
    - (nykyhetki - d(aika) $\leq$ d(abskesto)
  - Suhteellinen aikaeheys
    - yhden tapahtuman alkioiden arvojen suhde toisiinsa
    - kaikille $d' \in R$, | d(aika) - d'(aika) | $\leq$ R(suhtkesto)

## Esimerkki

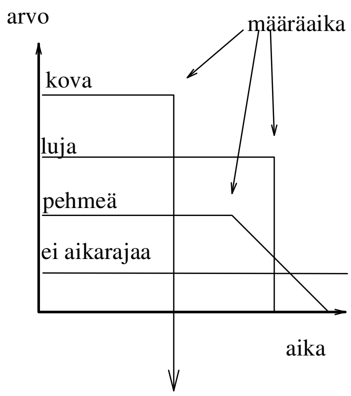
- R(suhtkesto) = 2, nykyhetki = 100
- lampotila=(347,5,95) ja paine=(50,10,97)
  - abs aikaeheä: (100-95) $\leq$ 5 ja (100-97) $\leq$ 10 ok
  - suht aikaeheä: | 95 - 97 | $\leq$ 2 ok
- lampotila=(347,5,95) ja paine=(50,10,92)
  - abs aikaeheä: (100-95) $\leq$ 5 ja (100-92) $\leq$ 10 ok
  - suht aikaeheä: | 95 - 92 | $\leq$ 2 EI

# Datan luokittelu




# Tosiaikatapahtuma ja aikarajat

- Kova (hard)
  - aikarajan ylitys katastrofi
- Luja (firm)
  - myöhästyneestä ei hyötyä
  - laskenta keskeytetään
- Pehmeä (soft)
  - myöhästyneestäkin iloa
  - hyöty putoaa ajan kuluessa



# Tosiaikatapahtuman ominaisuuksia

- Tärkeys (criticality)
  - miten välttämätöntä juuri tämän suoritus on
- Määräaika (deadline)
  - koska viimeistään tehty
- Aloitushetki (start time)
  - milloin liikkeelle
- Arvioitu kesto
  - suorituksen tarvitseman laskenta-ajan arvio

$$V_T(t) = \begin{cases} c_T & , s_T<t<d_T \\ c_T * (z_T-t)/(z_T-d_T), & d_T<t<z_T \\ 0 & , \text{muutoin} \end{cases}$$

missä
t - nykyhetki
$s_T$ - tapahtuman aloitushetki
$c_T$ - tapahtuman tärkeys 1 .. $C_{Tmax}$
$c_{Tmax}$ - maksimitärkeys
zT - aika, jolloin arvo muuttuu 0

J. Huang, etal, Real-Time Transaction processing: Design, Implementation and Perforaance Evaluation. Tekninen raportti COINS 90-43. University of Machachusetts, Toukokuu 1990. Myös teoksessaStankovic & Ramamrithmam. Advances in Real-Time systems.IEEE, 1993. s..587-621

# Skedulointi ja samanaikaisuudenhallinta

Skedulointi

- Jaksollisia vs. satunnaisia
- Rate monotonic
- Earliest deadline first
- Least slack
- High priority

Samanaikaisuudenhallinta

- Lukitseva vs. optimistinen
- 2PL-High priority
- 2PL-abort
- WAIT-X
- Time interval

# Esimerkki

| Tapahtuma | Saap.aika | kesto | määräaika | Data |
|---|---|---|---|---|
| A | 0 | 2 | 7,5 | X |
| B | 1 | 2 | 4 | X |
| C | 2 | 3 | 7 | Y |

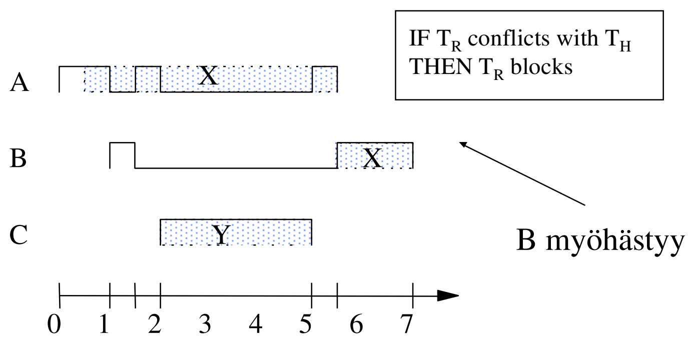
Puhdas EDF:

A B B C C C A

0 1 2 3 4 5 6 7

Abbott & Garcia-Molina, Scheduling Real-Time Transactions: A Performance Evaluation. ACM Tr on Database Systems. Vol 17, No 3, Sep 1992, s. 513-560

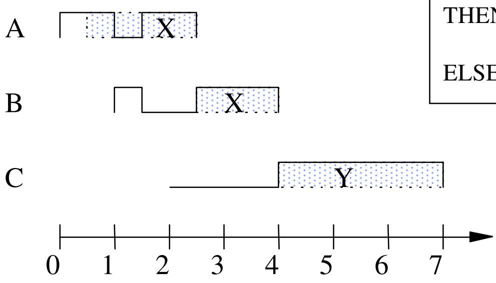
# Puhdas odotus (kuten 2PL)




Prioriteeteista riippumatta odota lukkoa, kunnes vapautuu

# Prioriteetin nosto

IF $P(T_R) > P(T_H)$
THEN $T_R$ blocks
$T_H$ inherit pri of $T_R$
ELSE $T_R$ blocks

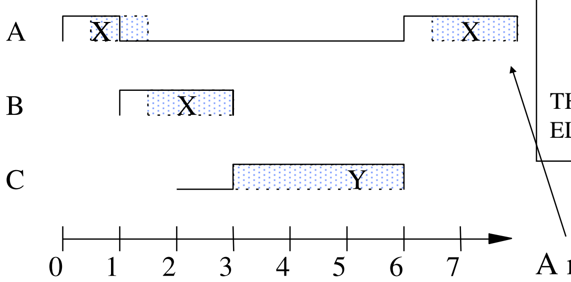



Kaikki ehtivät

Nosta lukonhaltijan prioriteetti odottajan prioriteetin tasolle

# Abortoi vähemmän tärkeä

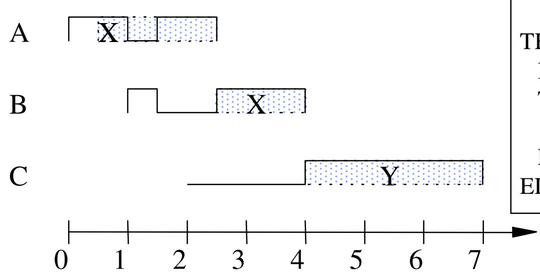
IF For all TH
holding a lock O
$P(T_R) > P(T_H)$ AND
$P(T_R) > P(T_H^A)$
THEN Abort each $T_H$
ELSE $T_R$ blocks




A myöhästyy hiukan (0,5)

Abortoi kaikki vähemmän tärkeät, joilla on haluttuja lukkoja

# Ehdollinen abortointi




IF $P(T_R) > P(T_H)$ AND
$P(T_R) > P(T_H{}^A)$
THEN
IF $S_R \geq E_H - P_H$
THEN $T_R$ blocks
$T_H$ inherit pri of $T_R$
ELSE Abort $T_H$
ELSE $T_R$ blocks

Jos on aikaa odottaa, niin nosta matalamman prioriteetti, muuten abortoi se

# Keskusmuisti vai levy

Keskusmuisti

- Nopea
- Tieto katoaa
- Pieni ( < 1 GB)
- Kallis
- Saantiaika lähes vakio

- Pysyvyys edellyttää säilyvää muistia tai levyä varmistukseen

Levy

- Hidas
- Pysyvä (ellei levy hajoa)
- Suuri ( > 100 GB)
- Halpa
- Saantiajassa suuria vaihteluja

- Saantiajan parempi ennakointi edellyttää prioris. noutoprotok.

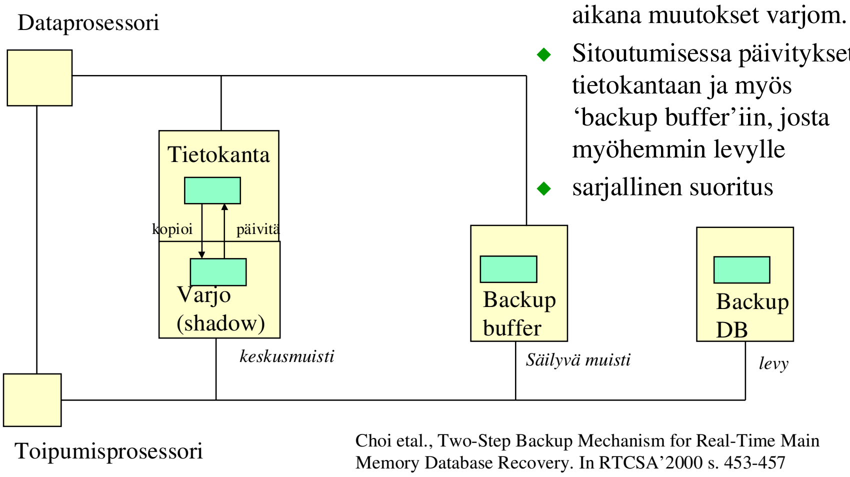
## Tiedon saatavuus ja vikasietoisuus

- Saatavuus
  - Kannan tiedot ovat käytettävissä x % ajasta
- Vikasietoisuus
  - Järjestelmä sietää ennaltamääritellyn määrän ennaltamääriteltyjä vikoja kaatumatta
- Ei suoranaisesti tosiaikaongelma, mutta erityinen ongelma tosiaikajärjestelmissä

## Tosiaikakannan vikasietoisuus

- *Saatavuuden* lisäys edellyttää monentamista
- *Levyvikoihin* varautuminen edellyttää monentamista (esim. RAID)
- Keskusmuistitiedon *pysyvyyden* takaaminen edellyttää levyä tai säilyvää muistia
- Toiminnan nopea *palauttaminen* (jos hetkellinen toimimattomuus sallitaan) edellyttää algoritmisia ratkaisuja tätä varten

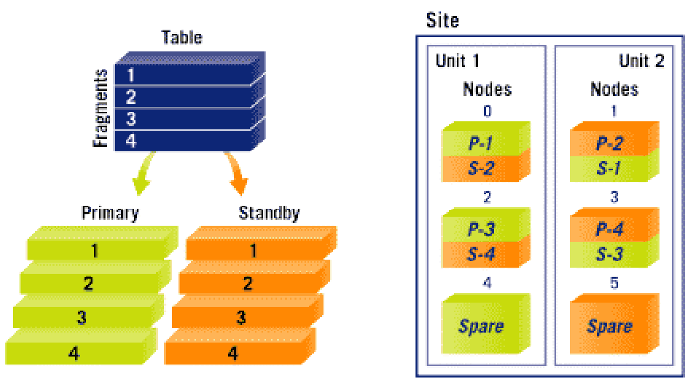
## Esimerkki arkkitehtuurista




- Tapahtuman suorituksen aikana muutokset varjom.
- Sitoutumisessa päivitykset tietokantaan ja myös 'backup buffer'iin, josta myöhemmin levylle
- sarjallinen suoritus

Choi etal., Two-Step Backup Mechanism for Real-Time Main Memory Database Recovery. In RTCSA'2000 s. 453-457

## Clustra




http://www.clustra.com/

# Yhteenveto

- Aikarajat ja tärkeys
- Suositaan tärkeitä ja tavoitellaan aikarajoja
- Keinoja:
  - keskusmuisti
  - rinnakkaisuuden hallinta ja skedulointi
  - monentaminen
- Tosiaikatietokanta vain jos
  - datalla on aikarajoja
  - tapahtumilla on aikarajoja