

# 58307301 Hajautetut algoritmit

seminaari syksyllä 2007  
vastuuhenkilö Jyrki Kivinen  
toinen vetäjä Timo Karvi

## Seminaarin suorittaminen

- kirjoitelma (10-15 sivua) 50%
- esitelmä (n. 45 min) 40%
- muu aktiivisuus (ml. kirjallinen palaute muille) 10%

Läsnäolopakko: kuunneltava vähintään 9 esitelmää (oman lisäksi); esitelmiä kaikkiaan 13.

## Toiminta periodilla I

- ei esitelmiä (poikkeus: 12.10.)
- 7.9. ensimmäinen kokoontuminen: seminaarin ja aihepiirin esittely
- 14.9. toinen kokoontuminen: aiheiden jakaminen, niihin tutustuminen
- 21.9. työsuunnitelman määräaika (lähdeluettelo, muutaman virkkeen hahmotelma sisällöstä)
- 12.10. luonnoksen määräaika (ainakin 5 sivua valmista tekstiä, loppu ranskalaisin viivoin)
- henkilökohtaista ohjausta saa pe 10-12 ja tarvittaessa muinakin aikoina (mutta joka tapauksessa sovi etukäteen sähköpostitse)
- yksinkertaisiin kysymyksiin saa vastauksia sähköpostitse

## Toiminta periodilla II

- ensimmäinen esitelmä 12.10. (I periodin viikko 6)
- loput esitelmät periodilla II
- 2 esitelmää / kokoontuminen
- esitelmä n. 45 min, mistä 5 min varataan keskustelulle
- kirjoitelman lopullisen version määräaika 29.10.

## Määräaikojen luonteesta

Työsuunnitelman ja kirjoitelmaluonnoksen tekemiselle voi saada (hieman) jatkoaikaa tekemällä

- perustellun hakemuksen
- **ennen** määräajan umpeutumista.

Perusteeton myöhästely alentaa arvosanaa.

Varsinaisen kirjoitelman määräaika on **tiukka**:

- mikä tahansa myöhästymisen alentaa arvosanaa
- myöhästymisen yli 3 vrk tai ilman ennakoilmoitusta johtaa hylkäämiseen.

## Materiaali

- Shlomi Dolev: *Self-Stabilization* (melkein kokonaan)
- Nancy Lynch: *Distributed Algorithms* (luvut 5 ja 6)
- muutama artikkelilähde

## Johdatus seminaarin aihepiiriin

Hajautettu laskenta: joukko prosessoreita rinnakkaisesti

- suorittaa paikallista laskentaa
- kommunikoi keskenään.

Huomio kiinnittyy järjestelmän

- vikasietoisuuteen
- kommunikoinnin määrään.

Erilaisia malleja:

- synkroninen vs. asynkroninen
- viestien lähettäminen vs. yhteinen muisti
- prosessorit samanlaisia / yksilöllisiä / yksi johtaja
- erilaiset virhemahdollisuudet

## Itsestabiloinnin perusajatus

Perusversiossa ajatellaan, että

- järjestelmä on toiminut jonkin aikaa mielivaltaisella tavalla virheellisesti
- erityisesti järjestelmä on voinut päätyä mihin tahansa "ristiriitaiseen" tilaan
- sitten virheet loppuvat.

Pystyykö järjestelmä palaamaan "konsistenttiin" tilaan (kun virheetön jakso kestää riittävän kauan)?

Eroaa malleista, joissa tehdään oletuksia virhemahdollisuuksista (esim. rajoitetaan virheellisten prosessorien lukumäärää):

- itsestabilointi on vahvempi: alkutila voi olla mielivaltainen
- itsestabilointi on heikompi: edellytetään täysin virheetön toipumisperiodi.

## Esimerkki itsestabiloinnista

- orkesteri soittaa ulkona
- tuuli yltyy ja kääntelee satunnaisesti soittajien nuotteja

⇒ orkesteri joutuu epätahtiin

- sitten tuuli lakkaa
- soittaja näkee, miltä sivuilta lähinaapurit soittavat
- pystyykö orkesteri synkronoimaan itsensä?

Menetelmä, joka ei toimi:

- jos enemmistö naapureista on keskenään samalla sivulla, seuraa heitä
- voi jäädä oskilloimaan.

Menetelmä, joka toimii:

- seuraa naapuria, joka on pieninumeroisimmalla sivulla.

## Klassinen esimerkki: keskinäinen poissulkeminen [Dijkstra 73]

- $n$  prosessoria  $P_i$ ,  $i = 1, \dots, n$ , joilla jokaisella oma rekisteri  $x_i$
- prosessorit kytketty renkaaseen  $P_1 \leftrightarrow P_2 \leftrightarrow P_3 \leftrightarrow \dots \leftrightarrow P_{n-1} \leftrightarrow P_n \leftrightarrow P_1$
- $P_i$ ,  $1 \leq i \leq n - 1$ , voi lukea myös rekisterin  $x_{i-1}$  (vasen naapuri);  
 $P_1$  voi lukea rekisterin  $x_n$
- **keskusdemoni** antaa suoritusvuoron yhdelle prosessorille kerrallaan (reilusti)
- suoritusvuoron saanut prosessori saa suorittaa häiriöttä jonon käskyjä
- **kierros** on ajanjakso, jonka kuluessa jokainen prosessori on saanut ainakin yhden suoritusvuoron.

## Protokolla keskinäiselle poissulkemiselle

Proessori  $P_1$  on erikoisasemassa: se suorittaa ohjelmaa

```
do forever
    if  $x_1 = x_n$  then  $x_1 := (x_1 + 1) \bmod (n + 1)$ .
```

Proessori  $P_i$ ,  $i = 2, \dots, n$ , suorittaa ohjelmaa

```
do forever
    if  $x_i \neq x_{i-1}$  then  $x_i := x_{i-1}$ .
```

Saadessaan suoritusvuoron keskusdemonilta prosessori suorittaa yhden iteraation **do**-silmukkaansa.

Jos jollain ajanhetkellä arvo  $x_i$  muuttuisi **jos**  $P_i$  saisi suoritusvuoron, sanomme että  $P_i$  **voisi muuttaa tilaansa** tällä ajanhetkellä

Tilannejono on **laillinen** keskinäisen poissulkemisen kannalta, jos sen aikana

- koskaan kaksi eri prosessoria eivät yhtä aikaa voisi muuttaa tilaansa
- jokainen prosessori voi muuttaa tilaansa äärettömän monta kertaa.

**Väite 1:** jos protokolla käynnistetään tilanteesta, jossa  $x_1 = x_2 = \dots = x_n$ , niin syntyvä tilannejono on laillinen.

**Väite 2:** jos protokolla käynnistetään mistä tahansa tilanteesta, niin  $x_1 = x_2 = \dots = x_n$  tulee voimaan  $O(n^2)$  kierroksen kuluessa.

Väitteet 1 ja 2 yhdessä implikoivat, että protokolla on itsestabiloituva.

Väitteen 1 totuus on melko ilmeinen. Tarkastellaan esim. tilannetta  $n = 4$  ja  $x_i = 3$  kaikilla  $i$ :

$x_1$	3	4	4	4	4	0	0	0	0	1	1	1	1	2	2	...
$x_2$	3	3	4	4	4	4	0	0	0	0	1	1	1	1	2	...
$x_3$	3	3	3	4	4	4	4	0	0	0	0	1	1	1	1	...
$x_4$	3	3	3	3	4	4	4	4	0	0	0	0	1	1	1	...

Sarakkeet eivät tyypillisesti edusta perättäisiä ajanhetkiä, koska välillä aktivoidaan prosessoreita jotka eivät voi muuttaa tilaansa.

Kuitenkin jokaisella kierroksella siirrytään ainakin yksi sarake eteenpäin.

Väitteen 2 todistamiseksi todetaan ensin, että  $x_1$  ei voi pysyä muuttumattomana  $n$  kierrosta.

**Vastaoletus:**  $x_1 = a$  yhtäjaksoisesti  $n$  kierrosta.

Kun yksi kierros on kulunut, pätee oletuksen mukaan edelleen  $x_1 = a$ . Koska  $P_2$  on ollut suorituksessa ainakin kerran, pätee myös  $x_2 = a$ .

Kun toinen kierros on kulunut, pätee  $x_1 = x_2 = x_3 = a$ .

Kun  $n - 1$  kierrosta on kulunut, pätee  $x_1 = x_2 = \dots = x_n = a$ .

Siis kierroksella  $n$  asetetaan  $x_1 := (a + 1) \bmod (n + 1)$ ; **ristiriita**.

Lähdetään nyt liikkeelle mielivaltaisesta tilanteesta  $c$ .

Olkoon  $j \in \{0, \dots, n\}$  sellainen, että tilanteessa  $c$  kaikille  $i$  pätee  $x_i \neq j$ . Ainakin yksi tällainen  $j$  on olemassa, sillä muuttujia  $x_i$  on  $n$  ja mahdollisia arvoja  $n + 1$  kappaletta.

Olkoon  $t$  tilanteen  $c$  jälkeen ensimmäinen ajanhetki, jolla arvo  $j$  tulee arvoksi jollekin  $x_i$ . Protokollasta nähdään suoraan, että tässä  $i = 1$ . Lisäksi  $t$  tulee vastaan korkeintaan  $n^2$  kierroksen jälkeen, sillä edellisen kalvon mukaan  $x_1$  saa uuden arvon joka  $n$ . kierros.

Hetken  $t$  jälkeen  $x_1$  säilyttää arvon  $j$ , kunnes  $x_n = j$ . Koska  $j$  ei voi tulla minkään  $x_i$ :n arvoksi "spontaanisti" vaan ainoastaan propagoitumalla muuttujasta  $x_1$  lähtien, tämä tapahtuu  $n$  kierroksen kuluessa, ja tällöin  $x_1 = x_2 = \dots = x_n = j$ .

Kaikkiaan kului korkeintaan  $n^2 + n = O(n^2)$  kierrosta.

## Aiheita esitelmille

Dolev luku 2: perustekniikoita, keskeisiä esimerkkejä

Dolev luku 3: mallin motivointia tietoliikenneprotokollien mahdollisilla virhetiloilla

Dolev luku 4: itsestabiloivan algoritmien muuntaminen laskentamallista toiseen (kolme esitelmää):

- yhteisestä muistista viestinvälitykseen
- yksilölliset prosessorit vs. yksi johtaja
- synkronointi

Dolev luku 5: tekniikoita ei-stabiloivan algoritmin muuttamiseksi stabiloivaksi

Dolev luku 6: itsestabiloinnin yhdistäminen erilaisiin vikamalleihin  
(proessorien "nukahtaminen"; bysanttilainen tapaus)

Dolev luku 7: stabilointi siten, että paikalliset virheet eivät sotke koko verkkoa

Beauquier, Gradinariu, Johnen: ylä- ja alarajoja muistintarpeelle itsestabiloivassa johtajan valinnassa

Chen, Welch: itsestabiloiva keskinäinen poissulkeminen ad hoc -verkossa

Johnen, Nguyen: ad hoc -verkon ryvästäminen itsestabiloivasti

Boulinier, Petit, Villain: tehokkaampi itsestabilointi kun rajoitteet ovat lokaaleja

Lynch luku 5: yksimielisyyden saavuttaminen kun viestejä voi kadota

Lynch luku 6: yksimielisyyden saavuttaminen kun joukossa voi olla salaliittolaisia (" bysanttilaiset kenraalit" )

Dalot, Dolev: sama itsestabiloivasti